Ubbo Visser

LNAI 3159

Intelligent Information Integration for the Semantic Web

Springer

VISIT…

Lecture Notes in Artificial Intelligence 3159

Edited by J. G. Carbonell and J. Siekmann

Subseries of Lecture Notes in Computer Science

This page intentionally left blank

Ubbo Visser

Intelligent Information Integration for the Semantic Web

Springer

eBook ISBN: 3-540-28636-5
Print ISBN: 3-540-22993-0

Created in the United States of America

Visit Springer's eBookstore at: http://ebooks.springerlink.com
and the Springer Global Website Online at: http://www.springeronline.com

Dedicated to my family Susan and Jannes as well as my parents
who always gave me support in the rough times...

This page intentionally left blank

Foreword

The Semantic Web offers new options for information processes. Dr. Visser is dealing with two core issues in this area: the integration of data on the semantic level and the problem of spatio-temporal representation and reasoning. He tackles existing research problems within the field of geographic information systems (GIS), the solutions of which are essential for an improved functionality of applications that make use of the Semantic Web (e.g., for heterogeneous digital maps). In addition, they are of fundamental significance for information sciences as such.

In an introductory overview of this field of research, he motivates the necessity for formal metadata for unstructured information in the World Wide Web. Without metadata, an efficient search on a semantic level will turn out to be impossible, above all if it is not only applied to a terminological level but also to spatial-temporal knowledge. In this context, the task of information integration is divided into syntactic, structural, and semantic integration, the last class by far the most difficult, above all with respect to contextual semantic heterogeneities.

A current overview of the state of the art in the field of information integration follows. Emphasis is put particularly on the representation of spatial and temporal aspects including the corresponding inference mechanisms, and also the special requirements on the Open GIS Consortium.

An approach is presented integrating information sources and providing temporal and spatial query mechanisms for GIS, i.e., the BUSTER system developed at the Center for Computing Technologies (TZI) which was defined according to the following requirements:

- Intelligent search
- Integration and/or translation of the data found
- Search and relevance for spatial terms or concepts
- Search and relevance for temporal terms

While distinguishing between the query phase and the acquisition phase, the above serves as the basis for the concept of the systems architecture. The

representation of semantic properties requires descriptions for metadata: this is where the introduced methods of the Dublin Core are considered, and it is demonstrated that the elements defined there do not meet with the requirements and consequently have to be extended.

Furthermore, important problems of terminological representation, terminological reasoning, and semantic translation are treated extensively. Again, the definition of requirements and a literature survey on the existing approaches (ontologies, description logics, inference components, and semantic translation) sets the scope. The chapter concludes with a comprehensive real-world example of semantic translation between GIS catalogue systems using ATKIS (official German catalogue) and CORINE (official European catalogue) illustrating the valuable functions of BUSTER.

Subsequently, the author attacks the core problems of spatial representation and spatial reasoning. The requirements list intuitive spatial denominations, place-names, gazetteers, and footprints, and he concludes that existing results are not expressive enough to enable the desired functionalities. Consequently, an overview of the formalisms of place-name structures is given which is based on tessellations and allows for an elegant solution of the problem through a representation with connection graphs, including an evaluation of spatial relevance. The theoretical background is explained using a well-illustrated example.

Finally, the requirements for temporal representations and the corresponding inference mechanisms are discussed. A qualitative calculus is developed which makes it possible to cover the temporal aspects which are also of importance to Semantic Web applications.

After the discussion of the set of requirements for an intelligent query system, the state of the BUSTER implementation is discussed. In a comprehensive demonstration of the system, terminological, spatial, and temporal queries, and some of their combinations are described.

An outlook on future research questions follows. In the bibliography, a good overview is given on the current state of the research questions dealt with.

This book combines in an exemplary manner the theoretical aspects of a combination of intelligent conceptual and spatio-temporal queries of heterogeneous information systems. Throughout the book, examples are provided using GIS functionality. However, the theoretical concept and the prototypical system are more general. The ideas can be applied to other application domains and have been demonstrated and tested, e.g., in the electronics and tourist domains. This demonstrates well that the approaches worked out are useful for practical applications – a valuable benefit for those readers who are looking for actual research results in the important areas of data transformation, the semantic representation of spatial and/or temporal relations, and for applications of metadata.

Bremen, May 2004 Otthein Herzog

Preface

When I first had the idea about the automatical transformation of data sets, which we now refer to as semantic translation, many of my colleagues were sceptical. I had to convince them, and when I showed up with a real-world example (ATKIS-CORINE) we founded the BUSTER group. This was in early 1999.

Since then, many people were involved in this project who helped with their critical questions, valuable suggestions, and ideas on how to develop the prototype. Two important people behind the early stages of the BUSTER idea are Heiner Stuckenschmidt and Holger Wache. I would like to thank them for their overview, their theoretical contributions, and their cooperation. I really enjoyed working with them and we hopefully will be able to do some joint work in the future again.

Thomas Vögele played an important role in the work that has been done around the spatial part of the system. His contributions in this area are crucial and we had fruitful discussions about the representation and reasoning components of the BUSTER system. At this point, I also would like to thank Christoph Schlieder, who gave me a thorough insight into the qualitative spatial representations and always contributed his ideas to our objectives. Some of them are now implemented in the BUSTER prototype.

The development and implementation of the system would not have been possible without people who are dedicated to programming. Most of the Master's students involved in our project were working on it for quite a long time. Sebastian Hübner, Gerhard Schuster, Ryco Meyer, and Carsten Krüwel were amongst the first "generation". I would like to thank them for their programming skills and patience when I asked them to have something ready as soon as possible. Sebastian Hübner now plays an important role in our project. Without him, the new temporal part of our system would be non-existent.

Bremen,
April 2004

Ubbo Visser

This page intentionally left blank

Table of Contents

Part I

Introduction and Related Work

This page intentionally left blank

1

Introduction

The Internet has provided us with a new dimension in terms of seeking and retrieving information for our various needs. Who would have thought about the vast amount of data that is currently available electronically ten years ago? When we look back and think about what made the Internet a success we think about physical networks, fast servers, and comfortable browsers, just to name a few. What one might not think about, a simple but important issue is the first version of HTML. This language allowed people to share their information in a simple but effective way. All of a sudden, people were able to define a HTML document and put their information piece on the Web. The given language was sloppy and almost anybody with a small amount of knowledge about syntax or simple programming use could define a web page. Even when language items such as end-tags or closing brackets were forgotten, the browser did the work and delivered the content without returning syntax errors. We believe this to be a crucial point when considering the success story of the Internet: give the people a simple but effective tool with the freedom to provide their information.

Providing information is one thing, searching and retrieving information is at least as important. Early browsers or search engines offered the opportunity to search for specific keywords, mostly searching for strings. The user was prompted with results in a rather simple way and had to choose the required information manually. The more data were added to the Web, the harder the search for information became. The latest versions of search engines such as Google provide a far more advanced search based on statistical evidences or smart context comparisons and rank the results accordingly. However, the users still have to choose the information they are interested in more or less manually.

Being able to provide data in a rather unstructured or semi-structured way is part of the problems with automatic information retrieval. This is the situation behind the activities of the W3C concerning the Semantic Web. The W3C defines the Semantic Web on their Web page as:

> "The Semantic Web is the abstract representation of data on the World Wide Web, based on the RDF standards and other standards to be defined. It is being developed by the W3C, in collaboration with a large number of researchers and industrial partners." [136][1]

The same page contains a definition of the Semantic Web that is of similar importance. This definition has been created by [8] and states

> "The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation." [136][2]

These definitions indicate the Web of tomorrow. If data have a well-defined meaning, engines will be able to intelligently seek, retrieve, and integrate information and generate new knowledge to answer complex queries.

The retrieval and integration of information is the focus of this paper. Before going into detail we would like to share some creative ideas, which can be a vision of what we can expect from the Semantic Web.

1.1 Semantic Web Vision

Bernes-Lee et al. [8] already gave us an insight of what we should be able to do with the help of data and engines working in the Web. In addition, the following can help to see where researchers want to arrive in the future. These ideas can be distinguished into four groups:

- **Short-term:** The following tasks are not far away from being solved or, are already solved to a certain extent.
  - *Being able to reply on an email via telephone call:* This requires communication abilities between a phone and an email client. Nowadays, the first solutions are available, however, vendors offer a complete solution with a phone and an email client that come in one package with more or less the same software. An example is the VoiceXML package from RoadNews[3]. The beauty of this point is that an arbitrary email client and an arbitrary phone can be used. The main subject is interoperability between address databases.
  - *Meaningful browsing support:* The idea behind this is that the browser is smart enough to detect the subject the user is looking for. If for instance, the user is looking for the program on television for a certain day on a web page, the browser could support the user by offering similar links to other web sites offering the same content.

[1] http://www.w3.org/2001/sw/, no pagination, verified on Oct 17, 2002.
[2] http://www.w3.org/2001/sw/, no pagination, verified on July 1st, 2003.
[3] http://www.roadnews.com, verified on July, 1st, 2003.

- **Mid-term:** These tasks are harder to solve and we believe that solutions will be available in the next few years.
  - *Planning appointments with colleagues by integrating diaries:* This is a problem already tackled by some researchers (e.g. [90]) and the first solutions are available. Pages can be parsed to elicit relevant information and through reference to published ontologies reasoning support, it is possible to provide user assistance. However, this task is not simple and many problems still have to be addressed. This task serves as one example of the ongoing Semantic Web Challenge (http://challenge.semanticweb.org).
  - *Context-aware applications:* Ubiquitous computing might serve as another keyword in this direction. Context-awareness (cf. [49]) has to deal with mobile computing, reduction of data, and useful abstraction (e.g., digital maps in an unknown city on a PDA).
  - *Giving restrictions for a trip and getting the schedule and the booking:* The scenario behind this is giving a computer the constraints for a vacation/trip. An agent is then supposed to check all the information available on the Web, including the local travel agencies and make the booking accordingly. Besides some severe technical problems, such as technical interoperability between agencies, we also have to deal with digital signatures and trust for the actual booking at this point. First approaches include modern travel portals such as DanCenter[4] where restrictions for a trip can be made and booking is also possible. This issue will be postponed for now.
- **Long-term:** Tasks in this group are again more difficult and the solutions might emerge only in the next decade.
  - *Information exchange between different devices:* Suppose, we are surfing the Web and see some movies we are interested in which will be shown on television during the next few days. Theoretically, we are able to directly take this information and program our VCR (e.g., WebTV[5]).
  - *Oral communication with the Semantic Web:* So far, plain commands can be given via speech software to a computer. This tasks goes even further: here, we think about the discussions of issues rather than plain commands. We also anticipate inferences and interaction.
  - *Lawn assistant:* Use satellite and weather information from the Web, background garden knowledge issued to program your personal lawn assistant.
- **Never:** Automatic fusion of large databases.

We can identify a number of difficult tasks that will most likely be difficult to solve. The automatic fusion of large databases is an example for this. On the other hand, we have already seen some solutions (or partly solutions) for

[4] http://www.dancenter.com, verified on July, 1st, 2003.
[5] http://about-the-web.com/shtml/WebTV.shtml, verified on June, 1st, 2003.

tasks that are grouped into short- and mid-term problems (e.g., integrating diaries). The following research topics can be identified with regard to theses ideas.

1.2 Research Topics

The research topics are as numerous as the problems. The number of areas discussed at the first two International Semantic Web Conferences in 2001/2002 [19, 60] can be seen as an indication of this. Some of the topics were: agents, information integration, mediation and storage, infrastructure and metadata, knowledge representation and reasoning, ontologies, and languages. These topics are more or less concerned with the development and implementation of new methods and technologies. Topics such as trust, growth and economic models, socio-cultural and collaborative aspects also belong to these general issues with regard to the Semantic Web and are concerned with other areas.

We will focus on some of the topics mentioned first: metadata and ontologies, or more general knowledge representation and reasoning with the help of annotated information sources. In general, we have to decide on an appropriate language to represent the knowledge we need. We have to bear in mind that this language has to be expressive enough to cover the necessary elements of the world we are modeling. On the other, hand we have to think about the people who are or will be using this language to represent and annotate their knowledge or information sources needed to be accessible via WWW. If we do not expect highly qualified knowledge engineers to do this job (which is unrealistic if we want to be successful with the Semantic Web) we need to compromise between the complexity and the simplicity of the language[6].

We will discuss how ontologies are used in the context of the Semantic Web in section 2. When we say 'ontology' we refer to Gruber's well-know definition [45], that an ontology is an explicit specification of a conceptualization. Please note that we do not focus on terminological ontologies only. The vision of the Semantic Web clearly reveals that also spatial information (e.g., location-based applications, spatial search) and temporal information (e.g., scheduling trips, booking vacations) will be needed. We will motivate our research interests with two important issues: firstly, how do we find information or better: can we improve nowadays search engines? Secondly, once we have found information, how do we integrate this information in our application? The next two sections give a brief overview about what has to be considered with regard to search and integration of information.

[6] This is an analogy to the growth of the "old" Internet. The simplicity of HTML was one of the keys for the success of the WWW. Almost everybody was able to create a simple Web page with some text and/or picture elements. There was no syntax check telling the user that there is a bracket open and he/she has to fix it. The browser showed a result and did forgive little mistakes. This sloppiness was important because it helped a vast amount of people (non-computer scientist) to use HTML.

1.3 Search on the Web

Seeking information on the Web is widely used and will become more important as the Web grows. Nowadays, search engines browse through the Web seeking given terms within web pages or text documents without using ontologies. Traditional search engines such as Yahoo are based on full-text search. These search engines are seeking documents, which contain certain terms. In order to give a more specific query, the user is often able to connect numerous terms with logical connectors such as AND, OR or NOT. The program extracts the text found from the documents and delivers the answer (usually a link to the found document) to the user. However, these search engines also use algorithms that are based on indexing for optimization purposes. The search engine then uses this index for seeking the answer. Yahoo has shown that this kind of search can be sufficient if the user knows what they are looking for. A clear disadvantage here is the fact that these search engines only search textual documents. Also, they have problems with synonyms, homonyms or a mistake while typing. These engines usually provide a huge amount of results that fulfill the requirement of the query, however, most of the results are not what the user intended.

Another type of search is the similarity-based search used in search engines such as Google. The engine is looking for documents, which contain text that is similar to a given text. This given text could be formulated by the user who is seeking the information or can be a document itself. The similarity is analyzed by the words used in the query and the evaluated documents. The engine usually uses homonyms and synonyms in order to get better results. The method extracts the text corpus out of the document and reduces it to a number of terms. A distance measure assigns the similarity to a numerical value between 0 and 1, where the similarity is the determined by the number of corresponding terms. The advantage of this kind of search is that there is no empty set of results and the results are ranked. A disadvantage is that only text documents can be used. Also, the similarity is based in given words and sometimes it is hard to find appropriate words for the search.

The main problem in these kinds of search is, that the amount of results are numerous. Also, most of the results are not accurate enough. The user has to know the terms they are looking for and cannot search within documents other that textual-based files and web pages. The reason for this is that uninformed search methods do not use background knowledge about certain domains.

Intelligent search methods take this into account and use additional knowledge to get better results. However, this requires a certain extent of modeling for the knowledge. The given documents are annotated with extra knowledge (metadata). The search can then be extended by search about the annotated metadata. This background knowledge can be employed for the formulation of the query by using ontologies and inference mechanisms. Also, the user can use this extra knowledge to generate abstract queries such as “all reports of the department X”. The reports can be project reports, reports about impor-

tant meetings, annual reports of the department etc. With ordinary search engines the user would have to ask more than once.

Intelligent search methods also include the classical way of search. The user will get more sophisticated results if he takes advantage of the additional knowledge. If the users do not know the exact terms they are looking for, they can also take advantage of the extra knowledge by using inference mechanisms of the ontology. However, this requires that the knowledge is formulated in a certain way and inference rules need to be available. The Semantic Web provides information with a well-defined meaning, and in the following we will use the term "search" for "intelligent search".

We have mentioned how intelligent search can help us to get better results. We have also explained that ontologies are the key to this. Seeking information with ontologies adds a new feature to the search process: we are able to use inference mechanisms in order to derive new knowledge. The search would even be more efficient if we would be able to integrate information from data sources. Integration in this context means that heterogenous information sources can be accessed and processed despite different data types, structures, and even semantics. The following subsection describes the integration tasks in more detail.

1.4 Integration Tasks

We distinguish different integration tasks that need to be solved in order to achieve complete integrated access to information, namely syntactic, structural, and semantic tasks.

Syntactic Integration

The typical task of syntactic data integration is to specify the information source on a syntactic level. This means, that different data type problems can be solved (e.g., *short int* vs. *int* and/or *long*). This first data abstraction is used to re-structure the information source. The standard technologies to overcome problems on this level are wrappers. Wrappers hide the internal data structure model of a data source and transform the contents to a uniform data structure model [143].

Structural Integration

The task of structural data integration is to re-format the data structures to a new homogeneous data structure. This can be done with the help of a formalism that is able to construct one specific information source out of numerous other information sources. This is a classical middleware task, which can be done with CORBA on a low level or rule-based mediators [143, 138]

on a higher level. Mediators provide flexible integration services of several information systems such as database management systems, GIS, or the World Wide Web. A mediator combines, integrates, and abstracts the information provided by the sources. Normally wrappers encapsulate the sources.

Over the last few years, numerous mediators have been developed. A popular example is the rule-driven TSIMMIS mediator [14, 89]. The rules in the mediator describe how information of the sources can be mapped to the integrated view. In simple cases, a rule mediator converts the information of the sources into information on the integrated view. The mediator uses the rules to split the query, which is formulated with respect to the integrated view, into several sub-queries for each source and combine the results according to query plan.

A mediator has to solve the same problems, which are discussed in the federated database research area, i.e., structural heterogeneity (schematic heterogeneity) and semantic heterogeneity (data heterogeneity) [68, 83, 67]. Structural heterogeneity means that different information systems store their data in different structures. Semantic heterogeneity considers the content and semantics of an information item. In rule-based mediators, rules are mainly designed in order to reconcile structural heterogeneity, whereas discovering semantic heterogeneity problems and their reconciliation play a subordinate role. But for the reconciliation of the semantic heterogeneity problems, the semantic level must also be considered. Contexts are one possibility to describe the semantic level. A context contains "metadata relating to its meaning, properties (such as its source, quality, and precision), and organization" [65]. A value has to be considered in its context and may be transformed into another context (so-called context transformation).

Semantic Integration

The semantic integration process is by far the most complicated process and presents us a real challenge. As with database integration, semantic heterogeneities are the main problems that have to be solved within spatial data integration [118]. Other authors from the GIS community call this problem inconsistencies [103]. Worboys & Deen [145] have identified two types of semantic heterogeneity in distributed geographic databases:

- Generic semantic heterogeneity: heterogeneity resulting from field- and object-based databases.
- Contextual semantic heterogeneity: heterogeneity based on different meanings of concepts and schemes.

The generic semantic heterogeneity is based on the different concepts of space or data models being used. The contextual semantic heterogeneity is based on different semantics of the local schemata. In order to discover semantic heterogeneities, a formal representation is needed.

Ontologies have been identified to be useful for the integration process [43]. Ontologies can be also be used to describe information sources. However, so far we have described the process of seeking *concepts.* If we look back to the vision of the Semantic Web described in section 1.1 we might also need use colloquial terms to search for *locations* (e.g., "Frankenwald", a forest area in Germany) and *time* (e.g., summer vacation 2003). If we combine these we might get a complex query seeking for a *concept@location in time,* e.g., *"Accommodation in Frankenwald during summer vacation 2003"*. We note that both the location and the time description are rather vague. Therefore, we need means to represent and reason about vague spatial and temporal information as well.

1.5 Organization

The next chapter gives an overview about existing approaches in the area of information integration covering the *terminological* part. Spatial and temporal information integration approaches with regard to the Semantic Web are nonexistent to our knowledge. However, we discuss the existing representation and reasoning approaches and their ability to support the needs of the Semantic Web. Chapter 3 gives a general introduction to and a conceptual overview about the BUSTER approach. The need for ontologies, the requirements for a system that deals with the query type *concept@location in time,* and a solution for the use of multiple ontologies will be discussed.

Chapter 4 describes our terminological approach. We have learned that formal ontologies can help to describe the meaning of concepts in a certain way. This is necessary if we would like to provide an automatic way to integrate or *translate* information sources. BUSTER offers this translation service also on the data level, which means that transformation rules from one context to another context can be generated and that then data sources can be transformed. We will discuss this and give an example of catalogue integration in the geographical domain.

Chapters 5 and 6 describe overviews of our approach with regard to spatial and temporal annotation, representation, and reasoning. These chapters follow the same structure: first, the requirements will be discussed. This leads to new representation schemes and reasoning abilities, which will be discussed next. A few words to the relevance factors, which are important to understand the results and the ranking of the results are also included. The chapters finish with an example.

Chapter 7 describes some implementation issues of the prototypical BUSTER system. It is a classical client/server system implemented in JAVA where the client can be either an browser-based applet or an application. A system demonstration is also included in this chapter. We describe simple terminological, spatial, and temporal queries and consider also possible com-

binations, leading to new types of queries. For instance, the triple combination leads us to the query type *concept@location in time*.

We conclude this paper discussing our approach(es) with regard to the requirements given in each chapter. Furthermore, we will outline some of the future work that needs to be considered in order to improve this line of research.

This overview paper discusses relevant topics that we have been published over the years. The publications in the appendix follow the topics mentioned above and describe our approaches in more detail. We will refer to these papers accordingly. However, the temporal part is new and has not been published yet.

This page intentionally left blank

2

Related Work

In this chapter, we will address several information integration approaches, which base on ontologies. The first section discusses approaches that only deal with problems in regards to the terminological search and integration. The remaining sections are devoted to related work that was completed in the area of qualitative spatial and temporal representation and reasoning.

2.1 Approaches for Terminological Representation and Reasoning

Due to the vast amount of information integration approaches that have been developed, it would be impossible to describe them all in detail within the scope of this overview. Therefore, the following discussion is restricted to conceptual levels of these approaches and their underlying ideas. The results described in this section have been published previously [141]. The evaluation of these approaches is shown following criteria that include the role of ontologies and the mappings that are used between ontologies and information sources and between multiple ontologies.

2.1.1 The Role of Ontologies

Initially, ontologies were introduced as an explicit specification of a conceptualization [45]. Therefore, ontologies may be used in an integration task to describe the semantics of the information sources and to make the content explicit. With respect to the integration of data sources, they may be used for the identification and association of semantically corresponding information concepts. Furthermore, in several projects ontologies take on additional tasks such as querying models and verification [3, 13].

Content Explication

In nearly all ontology-based integration approaches ontologies are used for the explicit description of the information source semantics. However, the way can differ in which the ontologies are employed. In general, three different directions can be identified: *single ontology approaches, multiple ontologies approaches* and *hybrid approaches* [141, 69][1]. The integration based on a single ontology seems to be the simplest approach because it can be simulated by the other approaches. Some approaches provide a general framework where all three architectures can be implemented (e.g., DWQ [12]). The following paragraphs give a brief overview of the three main ontology architectures and some important approaches that represent them.

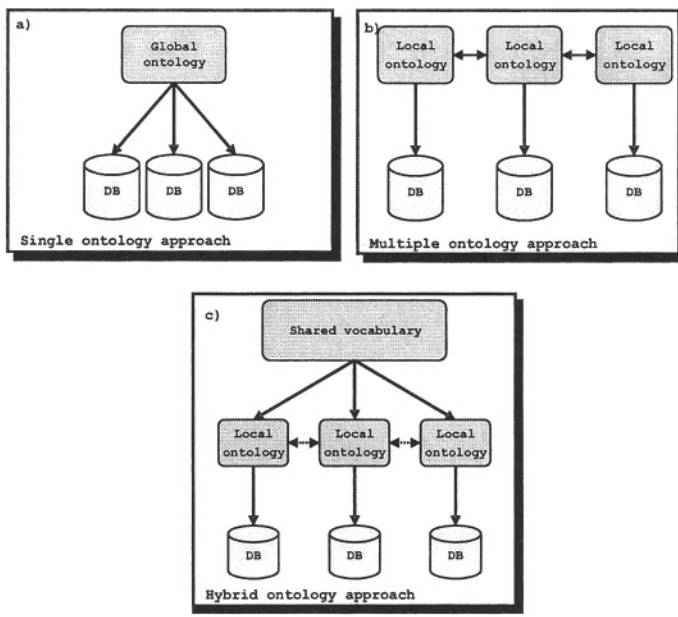


Fig. 2.1. Three ontology-based approaches.

Single Ontology Approaches

Single Ontology approaches (figure 2.1a) use one global ontology that provides a shared vocabulary for the specification of semantics. All information sources are related to one global ontology. The ontology describes the concepts of

[1] Klein [69] uses the terms *'merging* approach' for single ontology approach, *'mapping* approach' for multiple ontology approach and *'translation* approach' for hybrid approach.

a domain, which occur in the information sources. The information pieces therein are associated with terms of the ontology. This term specifies the semantic of the information piece.

Literature reveals that integration approaches using this idea are quite frequent [18, 73, 71, 40]. Among these are prominent approaches like SIMS [3]. The model of the application domain includes a hierarchical terminological knowledge base. Each source is simply related to the global domain ontology, i.e., elements of the structural information source are projected onto elements of the ontology. Users query the system using terms of the ontology. The SIMS mediator component reformulates this into sub-queries for the information sources.

Ontobroker [32] is another important representative of this group. An ontology is used here to annotate web pages with metadata. One can argue that the metadata comprise the knowledge contained on the web page, however, in a more formal and compact way. On this basis, users are able to locate web pages using ontological terms within their query.

The global ontology can also be a combination of several specialized ontologies. A reason for a combination of several ontologies can be the modularization of a potential large monolithic ontology. The combination is supported by ontology representation formalisms i.e., importing other ontology modules (cf. ONTOLINGUA [44]).

Single ontology approaches can be applied to integration problems where all information sources to be integrated provide nearly the same view on a domain. But, if one information source has a different view on a domain, e.g., by providing another level of granularity, finding the minimal ontology commitment [45] becomes a difficult task. Further, single ontology approaches are susceptible to changes in the information sources which can affect the conceptualization of the domain represented in the ontology. These disadvantages led to the development of multiple ontology approaches.

Multiple Ontology Approaches

In multiple ontology approaches (figure 2.1b), each information source is described by its own ontology. Studying the literature reveals that there are some systems following this approach, but considerably less than the single ontology approaches [81, 92, 12]. OBSERVER [81] is a prominent example of this group, where the semantics of an information source are described by a separate (source) ontology. In principle, this source ontology can be a combination of several other ontologies, but it can not be assumed, that the different source ontologies share the same vocabulary.

Therefore, multiple ontology approaches are those which use an ontology for each information source where the ontologies *differ* in their vocabulary. The advantage of multiple ontology approaches is that no common and minimal ontology commitment about one global ontology is needed [45]. Each source ontology can be developed without respect to other sources or their

ontologies. This ontology architecture can simplify the integration task and supports the change, i.e., the adding and removing of sources. On the other hand, the lack of a common vocabulary makes it difficult to compare different source ontologies. To overcome this problem, an additional representation formalism defining the inter-ontology mapping is needed.

The problem of mapping different ontologies is a well known problem in knowledge engineering. We will not try to review all the research that is conducted in this area but rather discuss general approaches that are used in information integration systems.

- *Defined Mappings:* a common approach to the ontology mapping problem is to provide the possibility to define mappings. This approach is taken in KRAFT [92], where translations between different ontologies are done by special mediator agents which can be customized to translate between different ontologies and even different languages. Different kinds of mappings are distinguished in this approach starting from simple one-to-one mappings between classes and values up to mappings between compound expressions. This approach allows a great flexibility, but fails to ensure a preservation of semantics: the user is free to define arbitrary mappings even if they do not make sense or produce conflicts.
- *Lexical Relations:* An attempt to provide at least intuitive semantics for mappings between concepts in different ontologies is made in the OBSERVER system [81]. The approach extends a common description logic model by quantified inter-ontology relationships borrowed from linguistics. In OBSERVER, relationships used are *synonym, hypemym, hyponym, overlap, covering* and *disjoint.* While these relations are similar to constructs used in description logics, they do not have a formal semantics. Consequently, the sub-sumption algorithm is rather heuristic than formally grounded.
- *Top-Level Grounding* In order to avoid a loss of semantics, one has to stay inside the formal representation language when defining mappings between different ontologies (e.g., DWQ [12]). A straightforward way to stay inside the formalism is to relate all ontologies used to a single top-level ontology. This can be done by inheriting concepts from a common top-level ontology and can be used to resolve conflicts and ambiguities (cf. [53]). While this approach allows connections to be established between concepts from different ontologies in terms of common super-classes, it does not establish a direct correspondence. This may lead to problems when exact matches are required.
- *Semantic Correspondences:* An approach that tries to overcome the ambiguity that arises from an indirect mapping of concepts via a top-level grounding and attempts to identify well-founded semantic correspondences between concepts from different ontologies. In order to avoid arbitrary mappings between concepts, these approaches have to rely on a common vocabulary for defining concepts across different ontologies. Wache

[137] uses semantic labels in order to compute correspondences between database fields. Stuckenschmidt et al. [108] build a description logic model of terms from different information sources and demonstrates that subsumption reasoning can be used to establish relations between different terminologies. Approaches using formal concept analysis (see above) also fall into this category, because they define concepts on the basis of a common vocabulary, to compute a common concept lattice.

The inter-ontology mapping identifies semantically corresponding terms of different source ontologies, e.g., which terms are semantically equal or similar. But the mapping also has to consider different views on a domain, e.g., different aggregation and granularity of the ontology concepts. We believe that in practice, inter-ontology mapping is very difficult to define.

Hybrid Approaches

To overcome the drawbacks of the single or multiple ontology approaches, hybrid approaches were developed (figure 2.1c). Similar to multiple ontology approaches the semantics of each source is described by its own ontology. In order to make the local ontologies comparable to each other they are built from a global shared vocabulary [41, 139, 138]. The shared vocabulary contains basic terms (the primitives) of a domain, which are combined in the local ontologies in order to describe more complex semantics.

In hybrid approaches an interesting point is how the local ontologies are described. In COIN [41] the local description of an information, so called context, is simply an attribute value vector. The terms for the context stems from a global domain ontology and the data itself. In MECOTA [139], each source concept is annotated by a label which combines the primitive terms from the shared vocabulary. The combination operators are similar to the operators known from the description logics, but are extended, e.g., by an operator which indicates that an information is an aggregation of several separated information pieces (e.g., a street name with number). Our BUSTER system uses the shared vocabulary as a (general) ontology, which covers all possible refinements, e.g., the general ontology defines the attribute value ranges of its concepts. A source ontology is one (partial) refinement of the general ontology, e.g., restricts the value range of some attributes. Because source ontologies only use the vocabulary of the general ontology, they remain comparable.

The advantage of a hybrid approach is that new sources can easily be added without modification. Also, it supports the acquisition and evolution of ontologies. The use of a shared vocabulary makes the source ontologies comparable and avoids the disadvantages of multiple ontology approaches. However, the drawback of hybrid approaches is that existing ontologies can not easily be reused. Instead, they have to be re-developed from scratch.

Other Ontology Roles

As stated above, ontologies are also used for a global query model or for the verification of a description formalized by a user or a system.

Query Model

The majority of the described integration approaches assume a global view (single ontology approach). Some of these approaches use the ontology as the global query scheme. SIMS [3] for one example: the user formulates a query in terms of the ontology. The system then reformulates the global query into sub-queries for each appropriate source, collects and combines the query results, and returns them thereafter.

Using an ontology as a query model has an advantage: the structure of the query model should be more intuitive for the user because it corresponds more to the user's understanding of the domain. However, from a database point of view, the ontology only acts as a global query scheme. If users formulate a query, they have to know the structure and the contents of the ontology. The user cannot formulate a query according to a scheme he would personally prefer. We therefore argue that it is questionable, whether the global ontology is an appropriate query model.

Verification

Several mappings must be specified from a global scheme to a local source schema during an integration process. The correctness of such mappings can be significantly improved if these can be verified automatically. A sub-query is correct with respect to a global query if the local sub-query provides a part of the queried answers, i.e., the sub-queries must be contained in the global query (query containment, cf.[12, 40]). Query containment means that the ontology concepts corresponding to the local sub-queries are contained in the ontology concepts related to the global query. Since an ontology contains a (complete) specification of the conceptualization, the mappings can be verified with respect to these ontologies.

In DWQ [12], each source is assumed to be a collection of relational tables. Each table is described in terms of its ontology with the help of conjunctive queries. A global query and the decomposed sub-queries can be unfolded to their ontology concepts. The sub-queries are correct, i.e., they are contained in the global query, if their ontology concepts are subsumed by the global ontology concepts. The PICSEL project [40] can also verify the mapping, but in contrast to DWQ, it can also generate mapping hypotheses automatically which are validated with respect to a global ontology.

The quality of the verification task strongly depends on the completeness of an ontology. If the ontology is incomplete, the verification result can erroneously imagine a correct query subsumption. Since in general the completeness can not be measured, it is impossible to make any statements about the quality of the verification.

2.1.2 Use of Mappings

The relation of an ontology to its environment plays an essential role in information integration. We already described inter-ontology mapping, which is also important to consider. Here, we use the term mappings to refer to the connection of an ontology to the underlying information sources. This is the most obvious application of mapping: to relate the ontologies to the actual contents of an information source. Ontologies may relate to the database scheme but also to single terms used in the database. Regardless of this distinction, we can observe different general methods used to establish a connection between ontologies and information sources.

- *Structure Resemblance:* a straightforward approach in connecting the ontology with the database scheme is to simply produce a one-to-one copy of the structure of the database and encode it in a language that makes automated reasoning possible. The integration is then performed on the copy of the model and can be easily tracked back to the original data. This approach is implemented in the SIMS mediator [3] and also by the TSIMMIS system [14].
- *Definition of Terms:* in order to clarify the semantics of terms in a database schema it is not sufficient to produce a copy of the schema. There are approaches such as BUSTER [114] that use the ontology to further define terms from the database or the database scheme. These definitions do not correspond to the structure of the database, they are only linked to the information by the term that is defined. The definition itself can consist of a set of rules defining the term. However in most cases, terms are described by concept definitions.
- *Structure Enrichment:* this is the most common approach in relating ontologies to information sources. It combines the two previously mentioned approaches. A logical model is built that resembles the structure of the information source and contains additional definitions of concepts. A detailed discussion of this kind of mapping is given in [64]. Systems that use structure enrichment for information integration are OBSERVER [81], KRAFT [92], PICSEL [40] and DWQ [12]. While OBSERVER uses description logics for both structure resemblance and additional definitions, PICSEL and DWQ define the structure of the information by (typed) horn rules. Additional definitions of concepts mentioned in these rules are achieved by a description logic model. KRAFT does not commit to a specific definition scheme.
- *Meta-Annotation:* another approach is the use of meta annotations that add semantic information to an information source. This approach is becoming prominent with the need to integrate information present in the World Wide Web, where annotation is a natural way of adding semantics. Approaches which are developed to be used on the World Wide Web are

Ontobroker [32] and SHOE [53]. We can further distinguish between annotations resembling parts of the real information and approaches avoiding redundancy. SHOE is an example of the former, Ontobroker of the latter.

2.2 Approaches for Spatial Representation and Reasoning

Space has many aspects and before we start describing existing approaches in this area, we would like to discuss the basics about the presentation of space. The following is mainly based on a paper presented by [17] who recently published an overview about this line of research.

The idea of spatial representation in general is to qualitatively abstract real objects of the world (i.e., discretize the world) in order to applying reasoning methods to compute queries such as "Which are the neighbors of region A?". It is also possible to give answers to this query with approaches purely based on quantitative models (GIS), however, there are strong arguments against this because these models are often intractable[2].

[34] argued that there is no pure qualitative spatial reasoning mechanism. Instead, a mixture of qualitative and quantitative information needs to be used to represent and reason about space. This is known as the 'poverty conjecture'. They also identified the property of transitivity of values as a key feature of qualitative quantitative spaces and conclude that operating with numbers will do proper reasoning. This leads to the challenge of the field of qualitative spatial reasoning (QSR): to provide calculi which allow the representation and reasoning of spatial entities without using traditional quantitative techniques.

Cohn and Hazarika state that since then (1987) a number of research approaches in the area of qualitative spatial representations emerged, which 'weakened' the poverty conjecture. Qualitative spatial representation addresses many aspects of space including ontology, topology, orientation, shape, size, and distance, just to name a few. The scope of this paper allows us to have a look at a few of these topics that are important to note for our main objectives with regard to the Semantic Web.

2.2.1 Spatial Representation

The first question is what kind of primitives of space should be used. This commitment to a particular ontology of space is not the only decision that has to be made when abstracting real-world objects with regard to spatial issues. Other decisions include the relationships between those spatial entities such as neighborhood, distances, shapes, etc. We discuss two main issues for our purpose, ontological and topological aspects.

[2] We might add that the use of quantitative spatial models also causes the user to compute a vast amount of data, which is not user-friendly for just querying the Web.

Ontological Aspects

The main point of discussion here concerns the spatial primitives. Traditionally, points are considered to be primary spatial entities (along with lines). An extension are regions which can be considered as sets of points. However, considering the reasoning issues, we can see a clear tendency of approaches which are in favor for regions as primitive spatial concepts [122].

Another ontological question concerns the nature of space which basically deals with the universe of the spatial entity. In other words: is the universe discrete or continuous? (cf. [80]). There are approaches that include either way, trying to find the connection between those two views. Galton [36] developed a high-level qualitative spatial theory based on a discrete model of space.

Further ontological questions involve the computational part of QSR. What kinds of basic operations are required or should be allowed with spatial primitives? The answer to this question depends on the needs for the application using the spatial models. Here, we also need to decide about the general approach, i.e., or do we represent our model symbolically or use another method, e.g., a graph-based approach. Either way, the underlying model together with the computational algorithms are sufficient enough to meet the demands given. This means that a number of necessary inference mechanism have to be provided.

Topology

The most important aspect of space is topology. Topological issues are fundamental for a number of qualitative spatial reasoning approaches since it is clear that topology can only be qualitative. Cohn and Hazarika argue that, although topology has been intensively studied in mathematical literature, only a few results are used to formalize common-sense spatial reasoning. One reason for this can be observed in the level of abstraction of the mathematical models (cf. also [42]).

For us, the main reason is clearly the focus of those mathematical theories. They usually deal with the representation of space rather then consider both representation and *reasoning* issues. To give an example, a typical spatial inference would be following: given that region a is in relation R_1 to region b and region b is in relation R_2 to region c. The reasoning engine would be able to prove what relations hold true for the regions a and c.

Some approaches adopt conventional mathematical formalisms [27, 144], others are based on axiomatic theories that can be found in the philosophical logic community [22, 142]. Most of these approaches, however, follow the 'pointless' geometry idea introduced by [38] where regions are taken as spatial primitives.

A prominent approach is the RCC calculus introduced by [93] (see also [26]). The idea behind the RCC is based on the connection of two regions a and b.

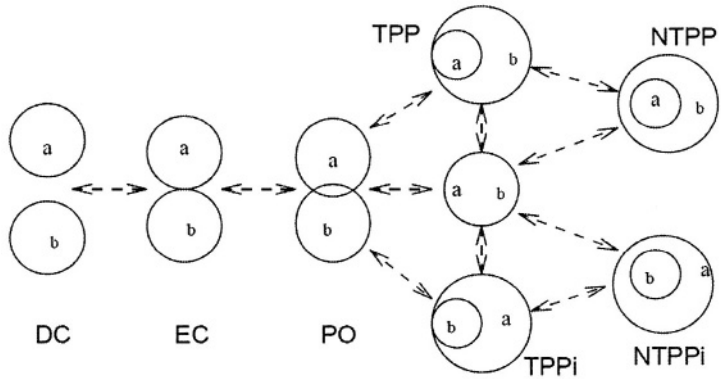


Fig. 2.2. RCC-8 relations, source: [17, p. 8]

The relation $C(a, b)$ (for connection) is more powerful that we might think. It is possible to define many predicates and functions that capture useful topological distinctions. Figure 2.2 shows the possible relations of the RCC-8 calculus and their continuous transitions.

The expressiveness of the RCC-8 has been thoroughly studied. $C(a, b)$ is expressive enough to define taxonomies of topological properties and relations. Other predicates can also be defined, one example is a predicate that counts the number of connections between two regions a and b. We will discuss this topic later in our spatial approach (section 5) and also give insight in the limitations of RCC-n in [99].

Further issues concerning the representation of space deal with extra information that is non-topological. One example is orientation, which cannot be determined with topological information only. We need additional information, e.g., in form of a point 'zero' or a *frame of reference*. This allows us to determine the orientation of a spatial object relatively to another object with regard to this frame of reference (or reference point).

Other points of interest are the distance between spatial objects and the size of a spatial object. The approaches can be distinguished by two groups: those using metrics and those using relative measurements. Details are discussed in [17].

2.2.2 Spatial Reasoning

In this section, we will restrict ourselves to reasoning components that are able to deal with static spatial information. The reason for this is twofold: first, the overall objective with regard to the Semantic Web suggests using static spatial knowledge and second, a thorough discussion of reasoning about spatial change would be beyond the scope of this paper.

The most prominent qualitative reasoning approaches include constraint-based reasoning. Here, the majority of the techniques are based on composition tables (cf. [55]). A compositional inference is a deduction where two relational facts of the form $R_1(a, b)$ and $R_2(b, c)$ are used to conclude another relational fact $R_3(a, c)$. Since compositional inferences do not depend on constants but on the logical properties of the relations, a lookup table can be generated and maintained with pairs of relations. This is of importance when dealing with a fixed set of relations. The composition table is usually $n \times n$, when n relations are given.

One can argue that the simplicity and effectiveness of the compositional inference technique makes it an attractive means for reasoning. This is emphasized by the numbers of researchers who are using this kind of inference mechanisms (e.g., [124, 27, 35]).

However, composition tables are not always the best choice due to complexity. Therefore, a good choice is then to use other, more general, constraint-based reasoning techniques. One example of this is to view QSR as a constraint-satisfaction problem [28]. Other approaches use theorem provers for their reasoning processes [6, 95], which will also be discussed in the next section.

2.2.3 More Approaches

By scanning the literature for spatial representation and reasoning approaches developed to especially serve the Semantic Web, one realizes that there are none[3]. However, spatial interoperability is a topic followed by a number of researchers in the areas of GIS and spatial reasoning (e.g., cf. [118] and the OpenGIS Consortium Specifications [88]). Another approach deals with qualitative spatial concepts and reasoning services based on description logics [82].

Open GIS Consortium Interoperability Program

OGC is an international industry consortium of more than 230 companies, government agencies and universities, participating in a consensus process to develop publicly available geo-processing specifications. Open interfaces and protocols that exist, support interoperable solutions that "geo-enable" the Web, including wireless and location-based services, and allow for complex spatial information and services accessible and useful with all kinds of applications.

The general approach within the OGC is to define specifications about geographical objects, protocols, services etc. Current initiatives include the Open Location Services Platform (OpenLS) [77]. This platform is also referred to as the GeoMobility Server (GMS). This server provides content such as maps,

[3] Not surprisingly, since the Semantic Web initiative is fairly new.

routes, addresses, points of interest, and traffic. It can also access other local content databases via the Internet. One of the core services provides access to an online directory to find the nearest or a specific place, product or service. Through a suitably equipped OpenLS application, the user starts to formulate the search parameters in the service request, identifying the place, product or service that they seek by entering the name, type, category, keyword, phone number, or some other 'user-friendly' identifier. A position must also be employed in the request when the subscribers seek the nearest place, product or service, or if they desire a place, product or service at a specific location or within a specific area. The position may be the current Mobile Terminal position, as determined through a Gateway Service, or a remote position determined in some other manner. The directory type may also be specified, e.g., yellow pages or a restaurant guide. Given the formulated request, the Directory Service searches the appropriate online directory to fulfill the request, finding the nearest or specific place, product or service depending on the search criteria. The service returns one or more responses to the query (with locations and complete descriptions of the place, product, or service, depending upon directory content), where the responses are ranked in order based upon the search criteria.

Use cases contain requests such as *"Where is the next Italian Restaurant?"* or *"Which Restaurants are within 1000m from my hotel?"*. In order to provide an answer to these type of questions (*concept@location*) the user has to be connected to spatial databases via the Internet. The databases contain OGC-defined polygons for locations and regions, which can be processed by the GMS. All locations are annotated with a defined coordinate system, namely the WGS 84 system (latitude, longitude).

This described approach is a new service, which is defined in XML for location-based services within the OpenLS Platform [77]. To date, this approach is still discussed in the community and has a good chance for recommendation through the OGC board.

Semantic-Based Information Retrieval

Möller et al. [82] investigated the use of conceptual descriptions based on description logic for content based information retrieval and presented an idea on how description logics can be extended with tools dealing with spatial concepts. They defined 15 topological relations that are organized in a subsumption hierarchy. In order to support spatial inferences, they extended CLASSIC [10] with new concept constructors based on the spatial relations. Their semantics assumes that each domain object is associated with its spatial representation (i.e., a polygon) via a predefined attribute *has-area.* Concepts for spatial objects are denoted with a predicate, a relation and a name for a polygon constant. They also have contributed to extending description logic theory by increasing the expressive power of description logics concerning reasoning about space (see also [46]). The Least Common Subsumer (LCS)

[15] operation has been extended in order to adequately deal with the spatial representation requirements for a TV-Assistant application. This proves that their theory works in practice.

2.3 Approaches for Temporal Representation and Reasoning

Before we start presenting a picture about existing approaches in this line of research, we would like to discuss the basics about the presentation of time. A profound source of this is the catalog of temporal theories, which has been written by [50]. The following is based in this compendium, except the summary of recent approaches.

Hayes introduces six meanings of time in his catalog of temporal theories. The first, and surely the most important one, sees time as a *physical dimension,* along with other physical dimensions such as voltage and length. The second meaning of time is what he called the *universe* of time, sometimes referred to as time line or time-plenum. The idea is that there is a endless discrete time stream. The third idea is based on pieces of time, also called *time-intervals.* An example of this is a time interval, which covers the rowing event at the last Olympic games. Another notion of time is that of a *point of time.* Here, we discuss a moment in the time continuum. While researchers still argue about the duration of a moment, we will postpone this discussion for now and go on to the fifth meaning of time: *duration.* An example of this is the amount of time needed to take a shower or get to work. The last notion of time is described as a position in a *temporal coordinate system,* such as June, 21st, 2003 or 5:15pm.

Hayes [50] argues that these time concepts have clear relationships to each other and can in fact be defined in various ways. Some theories follow the idea of taking time points as primitives, others are based on time intervals. The relation between points and intervals is important for the following, hence, we discuss this in more detail.

One view is that intervals *are* time points. These intervals are obviously as short as possible and thus, do not contain any sub-intervals (which is usually possible). They cannot overlay each other and do not have an internal structure. A colloquial term for this is the concept *moment.*

Another view is that there is an time continuum. This implies, that there is no such thing as a moment. The idea behind this is described in [2], who also illustrates the problem of meeting intervals. If two intervals meet, which interval “inherits” the meeting point? In fact, is it possible at all to decide whether a point belongs to the first or second interval? This is a relevant topic, since a number of temporal approaches are based on points as primitive objects. These approaches further define intervals as a set of points. The other view is to use points to locate positions in or between intervals, which themselves are primitive objects.

Hayes [50] concludes that it is impossible to divide an interval exactly symmetrically in half following the first notion of time. This implies that there must be open and closed intervals. The second intuition does allow this, however, rejects the conclusion that the meeting (or split) point is contained in either half.

Language Expressiveness

When describing time concepts, various *languages* can be used. These languages must cover temporal relations, allow propositions whose truth values might vary, and describe concepts whose properties might change over time.

One way to describe time is to use the concepts time themselves as objects. These objects can then be used in axioms depicting time to other things. An example for this is the following:

(submitted Ubbo_Visser PhD-Thesis 1995)

Another way to describe time ensures that sentences are 'true' at certain times. The following sentence states that it is true that I held a lecture on Artificial Intelligence 1 in Fall 2002.

(is_true (has_lecture Ubbo_Visser Artificial_Intelligence_1) Fall_2002)

Some theories use tenses. Tense logics extent usual logics by modal operators which allow to state that certain relations hold true in the past or in the future. Here is an example describing that I received my doctorate some time in the past (without saying when exactly).

(Past (has_received Ubbo_Visser Doctorate))

The final consideration with respect to language are temporal knowledge bases. The key behind this is that a language is imbedded in a temporal framework allowing to keep track of changes in the world and drawing inferences. The main problem here is to ensure consistency with the environment changing.

Following, we will give an overview about time point-based theories and interval-based theories. This subsection is partly based on [52].

2.3.1 Temporal Theories Based on Time Points

The temporal theories used in the approaches that we describe in the following are mostly consistent with the ideas stated by [50, p. 13]. A time interval is a piece of the time line, has a unique temporal extent, consists of two end points and is uniquely determined by these. Also, a time point can be uniquely determined by the extent of the interval between this point and some temporal position which we call 'zero'.

However, it is also possible to use other structures, which also rely on time points. Using computers implies some restrictions on the temporal theory. In order to distinguish between variations of time point structures (discrete vs. continuous, bounded vs. unbounded, linear vs. branched), we need to define the used terms.

Therefore, the elementary time points and the existing precedence relation $\prec$ are formalized. This relation is partially ordered, hence, transitivity (2.1) and irreflexivity (2.2) hold true.

$$\forall x, y, z[(x \prec y \wedge y \prec z) \rightarrow (x \prec z)] \tag{2.1}$$

$$\forall x \neg(x \prec x) \tag{2.2}$$

A time point structure is therefore an ordered pair $\langle X, \prec \rangle$ based on a non-empty set of time points X and a precedence relation $\prec$.

The mentioned variations, which are based on point structures, can be defined through axioms. Whether the time is bounded or not, for instance, is dependent on the existence or non-existence of a start or end point (2.3-2.6). A combination (restricted or bounded in one direction only) is also possible and can be useful.

$$\exists x_a \neg \exists x(x \prec x_a) \tag{2.3}$$

$$\exists x_e \neg \exists x(x_e \prec x) \tag{2.4}$$

$$\forall x \exists x'(x' \prec x) \tag{2.5}$$

$$\forall x \exists x'(x \prec x') \tag{2.6}$$

A discrete time model allows us to determine the direct neighbors on both sides of a non-marginal point (2.7,2.8). This model is isomorphic to natural numbers **N**. A dense time, on the other hand, is isomorphic to the rationals **Q** – where another point exists between pairwise disjunct time points (2.9)(cf. [50, p. 17]).

$$\forall x_1 [\exists x_2(x_2 \prec x_1) \rightarrow \exists x_3(x_3 \prec x_1 \wedge \neg \exists x_4(x_3 \prec x_4 \wedge x_4 \prec x_1))] \tag{2.7}$$

$$\forall x_1 [\exists x_2(x_1 \prec x_2) \rightarrow \exists x_3(x_1 \prec x_3 \wedge \neg \exists x_4(x_1 \prec x_4 \wedge x_4 \prec x_3))] \tag{2.8}$$

$$\forall x_l x_r [x_l \prec x_r) \rightarrow \exists x_m(x_l \prec x_m \wedge x_m \prec x_r)] \tag{2.9}$$

The notion of a one-dimensional, deterministic time line is described with the ordering axiom (2.10). There are no branches and the time points are totally ordered.

$$\forall x\, x' \,(x \prec x' \vee x = x' \vee x' \prec x) \tag{2.10}$$

Another notion is the one with a branching tree in one direction (e.g., future 2.11) Here, we only can compare time points if they are directly on the time line without being in the branch. The idea behind this is the indeterminism of potential future (or past) situations that can take place from the actual situation.

$$\forall xyz\,[(y \prec x \wedge z \prec x) \rightarrow (y \prec z \vee y = z \vee z \prec y)] \tag{2.11}$$

Point structures are therefore a model whose properties can be mathematically exactly defined.

2.3.2 Temporal Theories Based on Intervals

Human beings tend to formulate time with the help of intervals. These time intervals to a certain extent have interval structures as their underlying models. It is not necessary to have intervals only with exact same lengths, however, they must be non-empty, which basically means that start and end point are not the same. Again, axioms can be used to define the properties of these structures. The precedence relation is also partially ordered, hence, transitivity (2.1) and irreflexivity (2.2) hold true. In addition, we need a part-of relation $\subseteq$, which includes the identity and is therefore not a real part-of relation. Hayes calls this relation *inclusion* that has the properties transitivity (2.12), reflexivity (2.13), and anti-symmetry (2.14).

$$\forall x, y, z[(x \subseteq y \wedge y \subseteq z) \rightarrow (x \subseteq z)] \tag{2.12}$$

$$\forall x\,(x \prec x) \tag{2.13}$$

$$\forall x, x'[(x \subseteq x' \wedge x' \subseteq x) \rightarrow (x = x')] \tag{2.14}$$

We can therefore define an interval structure with the ordered triple $\langle X, \subseteq, \prec \rangle$, with the interval X, the inclusion $\subseteq$, and the precedence $\prec$.

Whether the time described by intervals is bounded or unbounded, dense, discrete, continuous etc. is similar to the properties of time point structures. However, the axiom describing *before* can be interpreted in different ways: a time interval (including end point) is fully before another time interval or it overlaps partially. This leads us to the definition of overlapping (2.15) which we can use to define the precedence relation (2.16).

$$\forall x, y[(x \cap y := \exists z(z \subseteq x \wedge z \subseteq y)] \tag{2.15}$$

$$\forall x, x'[(x \prec x') \rightarrow \neg(x \cap x')] \tag{2.16}$$

We can now transform the axioms 2.3 and 2.4 (earlier/later time point exists) and the axioms 2.5 and 2.6 (earlier/later time point do not exist) to interval structures. Because overlapping includes identity, we can define the ordering relation according to axiom 2.9, using $\cap$ instead of $=$.

$$\forall x\, x'\,(x \prec x' \vee x \cap x' \vee x' \prec x) \tag{2.17}$$

Considering the density or discreteness of the time model we have to take into account that intervals can include other intervals (inclusion) but no gaps. The latter needs another axiom which can be described as convexity axiom (2.18).

$$\forall x, y, z[(x \prec y \wedge y \prec z) \rightarrow \forall z'[(z' \subseteq x \wedge z' \subseteq z) \rightarrow (z' \subseteq y)]] \quad (2.18)$$

In summary, we can derive two demands with regard to the model: intervals can be infinitely divided into smaller intervals (time line is dense or continuous) or we have to deal with small but non-dividable intervals.

We can see that properties of time point structures and time interval structures can be described with similar axioms.

2.3.3 Summary of Recent Approaches

Temporal representation and reasoning is an essential feature in any activities that involve changes. This explains, why temporal representation and reasoning services are so important and appear in so many areas, including planning, natural language understanding, and knowledge representation.

Recent articles describe approaches in the area of *Temporal Constraint Programming,* an important area of temporal reasoning [102, 37]. Gennari describes a temporal reasoning system as a temporal knowledge base. It also contains a procedure to check its consistency, and inference mechanisms, which are able to derive new information and get a solution or all solutions to queries. Temporal reasoning tasks are mainly formulated as constraint satisfaction problems; therefore, the constraint satisfaction techniques can be used to check consistency, to search for solutions or all solutions for the given problem.

Events are the primitive entities in the knowledge base. They are characterized in temporal constraint programming by means of their time of occurrence, which can be given by time points or intervals (see above).

Temporal information can constrain events to happen at a particular time (e.g., "Coffee time is at 3:30 pm") or to hold during a time interval (e.g., "A class lasts 90 minutes"); moreover it can state relations between events of a qualitative type (e.g., "$Event_1$ is before $Event_2$") or of a metric one (e.g., "$Event_1$ has started at least three hours before $Event_2$").

Constraints can be either extensionally characterized by real or rational numbers, or intensionally represented as (finite) sets or relations of some algebra (e.g., Allen's interval algebra [2]). According to the formalization of constraints and the time unit chosen, the approaches can be classified into three main streams[4]:

- *Temporal reasoning with metric information:* In the quantitative approach to temporal reasoning with constraints, variables $X_1, \ldots, Xn$ range over real or rational numbers. Originally finite sets of real intervals, constraints are lately represented by unions of interval-sets. A temporal constraint is explicitly given as a set of intervals $I_1 \cup \ldots \cup I_n$ where $I_i = [l_i, r_i]$. The

[4] Other authors such as [102] and [123] describe these three main streams as *metric point* (for metric information), *qualitative point* and *qualitative interval* (for qualitative approaches based on Allen's interval algebra), and combinations (for mixed approaches).

constraints can be unary or binary and are represented by $\{I_1, \ldots, I_n\} = \{[l_1, r_1], \ldots, [l_n, r_n]\}$. An unary constraint T_i restricts the domain of a variable X_i to the given set of intervals. Thus, it is represented by the disjunction $(l_1 \leq X_i \leq r_1) \vee \ldots \vee (l_n \leq X_i \leq r_n)$. The binary constraint T_{ij} restricts the values for the distance of the variables $X_j - X_i$ and represents the disjunction $(l_1 \leq x_j - x_i \leq r_1) \vee \ldots \vee (l_n \leq x_j - x_i \leq r_n)$ [23]. The authors assume that all the intervals are pairwise disjoint.

Constraint propagation algorithms are based on metric properties of the continuous variable domain. Since the satisfiability problem of general temporal constraints is NP-hard, research if focussed on particular classes of temporal constraint problems such as single temporal constraint problems, backtracking algorithms, and constraint propagation algorithms in order to achieve local consistency or at least a good approximation of local consistency (e.g., [101]).

In principle, these methods can be used for reasoning services on the Semantic Web. However, the adaptation for their use implies a large modeling effort.

- *Qualitative approaches based on Allen's interval algebra:* The most fundamental and well-known theory about reasoning with time intervals has been formulated by [2]. This approach has been revised over the years and is based on interval structures, which are used as primitives.[5]

  Allen motivates his approach with the problem that much of our temporal knowledge is *relative,* and hence cannot be described by a date (or even a fuzzy date). As Allen further argues in his paper, his framework is particularly designed for these reasons:
  - it allows "significant imprecision": much temporal knowledge is relative and sometimes it has no relation to absolute dates;
  - "uncertainty of information" can be represented by means of disjunctions of relations between two intervals;
  - because of the qualitative representation of constraints one has a certain freedom when modeling knowledge and can choose the grain of reasoning, for instance expressing time in terms of days, weeks or business-days;
  - the reasoning engine allows for *default reasoning* of the type "If I parked my car in lot *A* this morning, then it should still be there now".

  In Allen's framework, variables range over real or rational valued intervals. Constraints are specified as unions of *atomic (basic)* relations, which are pairwise disjoint. Variables represent time intervals and the basic temporal relations are

$$Temporal\ relations = \left\{ \begin{array}{c} before, after, meets, met\_by \\ overlaps, overlaps\_by, during, contains, equals \\ starts, started\_by, finishes, finished\_by \end{array} \right\}$$

[5] There is a difference to the intervals described above since those intervals are composed by time points. Here, time intervals are primitives.

The class of all possible unions of the atomic relations forms a boolean algebra, Allen's interval algebra. There are 13 atomic relations and thus 2^{13} relations in total. Checking consistency for this algebra turned out to be NP-hard. Allen introduces a path-consistency algorithm to deal with the problems that propagates relations between intervals by means of composition. The algebra consists of $2^{13} = 8192$ relations which means that there are 2^{8192} possible subsets in that algebra, which make them intractable. Therefore, research in that area is concentrating on *tractable* and recently *maximal tractable* subalgebras. Some of the most important subalgebras of Allen's interval algebra are obtained by "translating" metric point relations into Allen relations. This means that there have to be languages to describe sets of qualitative or quantitative relations between points, and that these have to be translated in tractable subalgebras.

An exhaustive search by computers is a key technique to prove the maximality of the algebras that up to now have been discovered; this machine case analysis was firstly introduced by [86]. A different approach to this problem in a geometric and not a logic apparatus, is given in Ligozat's work [75, 74]. Some of the studied subalgebras are the Point Algebra [124, 5] and the NB algebra [86]. To compute a solution, backtracking search is used. It has been shown that the search gets more effective with the additional use of path-consistency checking such as a forward-checking method within the backtracking algorithm [102].

These mentioned arguments hold true also for the Semantic Web. Thus, interval-based approaches are valuable when discussing methods and techniques for temporal reasoning on the Web.

- *Mixed approach based on metric and qualitative constraints:* In this framework, the other approaches are combined in order to gain expressiveness, while trying not to loose the tractability of the problem; however, the complexity results are not always optimal. The ontological entities in the first approach are time points only, and the primitive entities in the second approach are time intervals. This third approach involves both points and intervals as primitive objects of the language; therefore new relations are introduced in order to "relate" time points and time intervals.

  Some authors have studied particular metric temporal constraint problems in order to find new sub-algebras of interval algebra. This can be seen as a qualitative approach because its main goal is an interval algebra. An approach is "mixed" when it aims at using both the expressive power of the qualitative and of the quantitative approaches to create "new" temporal frameworks, of which the satisfiability can be decided in polynomial time. The research in this direction is one of the most promising [107], however, the relative literature is still scarce.

2.4 Evaluation of Approaches

After discussing the approaches in these three areas, we need to verify further, whether they are suitable for the general needs and requirements mention in the introduction. Given that some of these ideas were introduced before the Semantic web emerged, we can conclude that some features and adaptations must be made. Please note we also discuss eligible approaches in chapters 4,5 and 6 accordingly. At this point we would like to discuss some major general issues.

2.4.1 Terminological Approaches

The Semantic Web demands some kind of formalization to ensure that engines are able to interpret information automatically. This important point must be generally taken into account. The following question arises: how do we formalize the knowledge? Naturally there are many ways to accomplish this, however, our survey in regards to intelligent information integration approaches [141] revealed that ontology-based approaches are the way to go.

The reasons for this statement are manifold and we would like to discuss a few. First, and probably the most important one is the activity in the working groups of the W3C. Both the Semantic Web and the ontology language working groups are close to achieving their goals: to create a common ontology language as a de facto standard to describe information on the Web. Interviews with two key players in this area, James Hendler and Patrick Hayes, revealed that most of the Web, not only the Semantic Web, is about defining standards that people can use and live with [54, 51].

Second, ontology-based approaches have a high degree of formality. They provide enough expressiveness (most, not all ontology languages are based on description logics) without losing decidability. This is crucial because people using the Semantic Web rely on this requirement.

Third, we need to be careful with metadata. If we want the Semantic Web to work, we need to ensure that the information contained on web pages, databases and multimedia documents are properly annotated. New professional applications such as Adobe Acrobat already support automatically the annotation of documents using RDF. Another demand for ontologies is that people should be able to use their own terms (they must be formalized ontologies). This is what we called intuitive labeling.

Fourth, we have observed lots of activity in ontology construction. A prominent example is the ontology of the US National Cancer Institute. This ontology consists of more than a million cancer terms with approximately 20.000 classes[6]. We can expect more ontologies in various areas over the next few years.

[6] http://www.mindswap.org/2003/CancerOntology, verified on June, 23rd, 2003

These issues lead us to the conclusion that using hybrid ontology approaches, some kind of description logics as ontology language should be supported by a reasoning engine available on the Web.

2.4.2 Spatial Approaches

Most of the spatial approaches are based on constraint-based reasoning methods. We have ruled out for now the changing spatial world and deal only with static knowledge in this area. We believe that in analogy to the terminological part, we need spatial ontologies to meet the requirement of the Semantic Web. The following statement can then be made:

There is a need for intuitive spatial names, especially for querying on the Semantic Web. Most people would like to use colloquial terms rather than cryptic terms or administrative concepts such as 'square 1234'. Perhaps this seems unimportant in the first place, however, if we want people to use the Semantic Web, we must provide them with acceptable solutions. Unfortunately, none of these approaches mentioned meets the demand. So therefore, we must develop a new method for intuitive labeling and construct spatial ontologies.

Another important issue is the data volume that is required to be processed over the Web. We know that metric-based approaches (GIS) are able to derive high quality knowledge. However, the main drawback behind using GIS or the OGC approaches are the vast amount of data that must be processed to answer a query. Spatial reasoning components are usually intersected with GIS and although this is probably fast enough, the information is not publicly available over the Semantic Web. The OGC also runs working activities dealing with models for billing these services. Therefore, the data process problems along with the fact that one must pay for these services leads us to believe that there is a need to develop a new spatial model with appropriate reasoning service.

2.4.3 Temporal Approaches

Most of the representation and reasoning approaches in this area are based on point or interval structures using either composition tables or constraint-based methods. Again, we believe, that, in analogy to the terminological and spatial part, temporal ontologies are needed to meet the requirements of the Semantic Web. The following statements underline this.

There is a need for intuitive temporal names, especially when people are involved querying the Semantic Web. As with spatial terms, people would like to use common words for temporal concepts such as 'Summer vacation 2003' rather than fill in a W3C temporal date format (cf. section 6.1). Further, none of the discussed approaches can meet this demand, therefore, we must develop new methods for this intuitive labeling and construct temporal ontologies.

The approaches that are based on temporal intervals are basically eligible for our purpose, however, the existing methods need an significant extension. One reason for this is that none of the approaches are able to express fuzzy boundaries. An example for a fuzzy boundary is the temporal concept ‘middle-age’. Experts argue about the exact time interval belonging to the Middle Ages, however, it is clear that the latest beginning of the Middle Ages is the reign of Karl the Great. Further, another clear disadvantage of the existing approaches is the lack of references to other intervals. It is not possible, e.g., to state that the earliest begin of the Middle Ages was the end of the Westroman Empire, which itself can be dated precisely. Therefore, there is a need to develop more sophisticated tools based on the previously mentioned approaches.

Part II

The Buster Approach for Terminological, Spatial, and Temporal Representation and Reasoning

This page intentionally left blank

3

General Approach of Buster

The general approach of BUSTER (Bremen University Semantic Translator for Enhanced Retrieval) follows the hybrid ontology approach described in section 2. This means that the overall architecture is based on annotated information sources, which are linked or can be found with an ontology-based retrieval mechanism. BUSTER can be looked at as a middleware, which can be used by various applications such as e-commerce programs, GIS-applications etc. It provides two subsystems, one acts as intelligent search engine, the other can be used for information integration or semantic translation. This general approach has been also discussed in [127, 133, 129, 130, 141, 132, 110].

3.1 Requirements

We consider the needs and future research lines of the Semantic Web community and can define the following overall requirements of the system:

Firstly, we would like to set up the system with an *intelligent search* component. This means, that the system should be able to find information sources with intelligent search methods described in section 1.3. *Integration* and/or *translation* of the found information is another important task, which includes the integration of information on the concept level and also an optional context transformation on the data level. This is just one of the essential needs for the Semantic Web to become successful. However, this implies that the data on the Web have to be annotated with background knowledge. Moreover, this background knowledge has to have some kind of formality to provide full support for both the retrieval and the integration process.

Secondly, the user should be able to query the Web with more than just the terms they are seeking. An important feature is the search for *spatial* terms or concepts. An example is someone looking for accommodation in a certain place. This place should also include colloquial terms rather than (x,y)-coordinates, which are used within monolithic GIS. Further, the inclusion of GIS into our approach would involve the download of a huge amount of data

given the fact that GIS "inferences" run on polygons. These polygons however, are usually of high resolution and therefore contain a lot of data. GIS are normally used for planning purposes and are run by official departments. We argue that many Semantic Web applications do not need data of this high resolution. In addition, using the GIS data on a more qualitative level would be of high value. One reason for this is the amount of data traffic (in terms in bytes) on the Web.

Another important aspect is the possibility to look for *temporal* terms on the Web. An extension of the former example clarifies this: we should be able to look for accommodation at certain places during a certain time. These temporal terms should also be colloquial such as "Easter vacation 2003". Currently, the W3C offers time specification with two exact time stamps. We will see later (section 6) that this does not fulfill the needs for the Semantic Web.

3.2 Conceptual Architecture

The BUSTER architecture provides an integrated solution for the problem of information retrieval and integration. We take into account all three levels of integration (cf section 1.4) combining several technologies including: standard markup languages, mediator systems, ontologies and knowledge-based classifiers. This holds particularly true for the terminological part of the BUSTER system, however, the overall approach of using ontologies and a common vocabulary also applies for the spatial and temporal part.

Figure 3.1 gives an overview about the BUSTER architecture. In general, the architecture can be divided in two distinct phases: an acquisition phase and a query phase.

During the acquisition phase, all desired information for providing a network of integrated information sources is acquired. This includes the acquisition of a Comprehensive Source Description (CSD) (see section 3.3 below) of each source together with the Integration Knowledge (IK), which describes how the information can be transformed from one source to another.

In the query phase, a user or an application (e.g., an e-commerce application, a GIS or a user searching for information on the Web) formulates a query which implies to an integrated view of sources. Several specialized components in the query phase use the acquired information, i.e., the CSD's and IK's, to select the desired data from several information sources and transform it into the structure and the context of the query.

All software components in both phases are associated to three levels: the syntactic, the structural and the semantic level. The components on each level deal with the corresponding heterogeneity problems. The components in the query phase are responsible for solving the corresponding heterogeneity problems whereas the components in the acquisition phase use the CSD's from

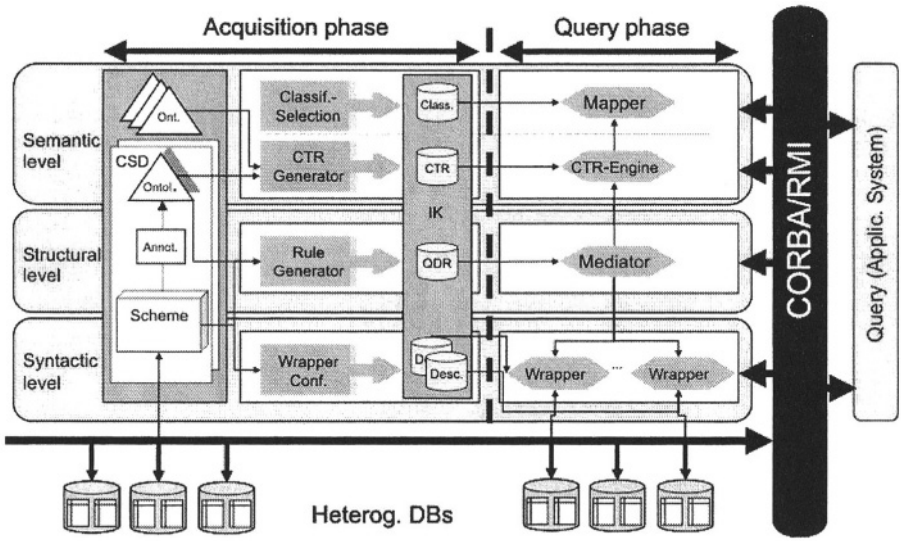


Fig. 3.1. A conceptual view of BUSTER.

the sources to provide the specific knowledge for the corresponding component in the query phase. A mediator for example, which is associated with the structural level, is responsible for the reconciliation of the structural heterogeneity problems. The mediator is configured by a set of rules that describe the structural transformation of data from one source to another. The rules are acquired in the acquisition phase with the help of the rule generator.

An important characteristic of the BUSTER architecture is the semantic level, where two different types of tools exists to solve the terminological semantic heterogeneity problems. This demonstrates one focus of the BUSTER system, to provide a solution for this type of problems. Furthermore, the need for two types of tools exhibits, that the reconciliation of semantic problems is very difficult and must be supported by a hybrid architecture where different components are combined.

In the following sections we describe the two phases and their components.

3.2.1 Query Phase

In the query phase, a user submits a query request for one or more data sources in the network of integrated data sources. In this phase several components of different levels interact.

On the syntactic level, *wrappers* are used to establish a communication channel to the data source(s), which is independent of specific file formats and system implementations. Each generic wrapper covers a specific file- or data-format. For example, generic wrappers may exist for ODBC data sources,

XML data files, or specific GIS formats. Still, these generic wrappers have to be configured to the specific requirements of a data source.

The mediator on the structural level uses information obtained from the wrappers and "combines, integrates and abstracts" [143] them. In the BUSTER approach, we use generic mediators which are configured by transformation rules (query definition rules QDR). These rules describe in a declarative style, how the data from several sources can be integrated and transformed to the data structure of the original source.

On the semantic level, we use two different tools specialized for solving the semantic heterogeneity problems. Both tools are responsible for the context transformation, i.e., transforming data from a source-context to a goal-context. There are several methods how the context transformation can be applied. In BUSTER we consider the context transformation by rules (also functional context transformation) and context transformation by re-classification [114, 138].

In the functional context transformation, the conversion of data is completed by applying predefined functions. A function is represented in Context Transformation Rules. These (CTR's) describe from which source-context to which goal-context the data can be transformed by the application of which function. The context transformation rules are invoked by the CTR-Engine. The functional context transformation can be used, e.g., for the transformation of area measures in hectares to area measures in acres, or the transformation of one coordinate system into another. All context transformation rules can be described with the help of mathematical functions.

Further to the functional context transformation, BUSTER also allows the classification of data into another context (Mapper in figure 3.1). This is utilized to automatically map the concepts of one data source to concepts of another data source. To be more precise, the context description (i.e., the ontological description of the data) is re-classified. The source-context description, to which the data is annotated, is obtained from the CSD, completed with the data information and relates to goal-context descriptions. After the context re-classification, the data is simply replaced with the data which is annotated with the related goal-context. Context re-classification together with the data replacement is useful for the transformation of catalog terms, e.g., exchanging the term of a source catalog by a term from the goal catalogue.

3.2.2 Acquisition Phase

Before the first query can be submitted, the knowledge, in fact the Comprehensive Source Description (CSD) and Integration Knowledge (IK) has to be acquired. The first step of the data acquisition phase consists of gathering information about the data source that is to be integrated. This information is stored in a source-specific data base, the CSD. A CSD has to be created for each data source that participates in a network of integrated data sources.

The Comprehensive Source Description

Each CSD consists of metadata that describe technical and administrative details of the data source as well as its structural and syntactic schema and annotations. In addition, the CSD comprises a source ontology, i.e., a detailed and computer-readable description of the concepts stored in the data source. The CSD is attached to the respective data source and should be available in a highly interchangeable format (e.g., XML), that allows for easy data exchange over computer networks.

Setting up a CSD is the task of the domain specialist, who is responsible for the creation and maintenance of the specific data source. With the help of specialized tools that use repositories of pre-existing general ontologies and terminologies, the tedious task of setting up a CSD can be supported. These tools examine existing CSD's of other but similar sources and generate hypotheses for similar parts of the new CSD's. The domain specialist must verify - eventually modifying - the hypotheses and add them to the CSD of the new source. With these acquisition tools the creation of new CSD's can be simplified.

The Integration Knowledge

In a second step of the data acquisition phase, the data source is added to the network of integrated data sources. In order for the new data sources to be able to exchange data with the other data sources in the network, Integration Knowledge (IK) must be acquired. The IK is stored in a centralized database that is part of the network of integrated data sources.

The IK consists of several separated parts, which provide specific knowledge for the components in the query phase. For example, the rule generator examines several CSD's and creates rules for the mediator (Wache et al., 1999). The wrapper configurator uses the information about the sources in order to adapt generic wrappers to the heterogeneous sources.

Creating the IK is the task of the person responsible for operating and maintaining the network of integrated data sources. Due to the complexity of the IK needed for the integration of multiple heterogeneous data sources and the unavoidable semantic ambiguities, it may not be possible to accomplish this task automatically. However, the acquisition of the IK can be supported by semi-automatic tools. In general, such acquisition tools use the information stored in the CSDs to pre-define parts of the IK and propose them to the human operator who makes the final decision about whether to accept, edit, or reject them.

It turned out that a proper annotation of information sources is crucial for our approach. Therefore, we introduce the comprehensive source description in more detail.

3.3 Comprehensive Source Description

In order to describe existing data metadata have to be used. Hence, we have to find an eligible language for the description. Over the last decade numerous meta data formats have emerged (e.g., Dublin Core, ISO/TC211). A good overview about existing meta information systems can be found in [132]. Since we are not dependent on any specific domain, in fact, we would like to use a general way to describe the data, we decided to use the Dublin Core Element Set, version 1.1 as a de facto basis for our CSD. The definitions utilize a formal standard for the description of metadata elements. The authors claim that the formalization helps to improve consistency with other metadata communities and enhances the clarity, scope, and internal consistency of the Dublin Core metadata element definitions [20].

3.3.1 The Dublin Core Elements

This section is based on the reference description of the Dublin Core Metadata Element Set. [63]. The current list of elements consists of 15 elements that have a descriptive name intended to convey a common semantic understanding of the element. The elements are given below:

1. *Title:* The name given to the resource, usually by the creator or publisher.
2. *Creator:* The person or organization primarily responsible for creating the intellectual content of the resource. For example: authors in the case of written documents, artists, photographers, or illustrators in the case of visual resources.
3. *Subject:* The topic of the resource. Typically, subject will be expressed as keywords or phrases that describe the subject or content of the resource. The use of controlled vocabularies and formal classification schemas is encouraged.
4. *Description:* A textual description of the content of the resource including abstracts in the case of document-like objects or content descriptions in the case of visual resources.
5. *Publisher:* The entity responsible for making the resource available in its present form, such as a publishing house, a university department, or a corporate entity.
6. *Contributor:* A person or organization not specified in a creator element, who has made significant intellectual contributions to the resource but, whose contribution is secondary to any person or organization specified in a creator element (for example, editor, transcriber, and illustrator).
7. *Date:* A date associated with the creation or availability of the resource. Recommended best practice is defined in a profile of ISO 8601 (http://www.w3.org/TR/NOTE-datetime) that includes (among others) dates of the forms YYYY and YYYY-MM-DD. In this scheme, the date 1994-11-05 corresponds to November 5, 1994.

8. *Type:* The category of the resource, such as home page, novel, poem, working paper, technical report, essay or dictionary. For the sake of interoperability, type should be selected from an enumerated list that is under development in the workshop series.
9. *Format:* The data format and optionally, dimensions (e.g., size, duration) of the resource. The format is used to identify the software and possibly hardware that might be needed to display or operate the resource. For the sake of interoperability, the format should be selected from an enumerated list that is currently under development in the workshop series.
10. *Identifier:* A string or number used to uniquely identify the resource. Examples for networked resources include URLs and URNs (when implemented). Other globally-unique identifiers, such as International Standard Book Numbers (ISBN), or other formal names would also be candidates for this element.
11. *Source:* Information about a second resource from which the present resource is derived. While it is generally recommended that elements contain information about the present resource only, this element may contain metadata for the second resource when it is considered important for discovery of the present resource.
12. *Language:* The language of the intellectual content of the resource. Recommended best practice is defined in RFC 1766 http://info.internet.isi.edu/in-notes/rfc/files/rfc1766.txt
13. *Relation:* An identifier of a second resource and its relationship to the present resource, this element is used to express linkages among related resources. For the sake of interoperability, relationships should be selected from an enumerated list that is currently under development in the workshop series.
14. *Coverage:* The spatial and/or temporal characteristics of the intellectual content of the resource. Spatial coverage refers to a physical region (e.g., celestial sector) using place names or coordinates (e.g., longitude and latitude). Temporal coverage refers to what the resource is about rather than when it was created or made available (the latter belonging in the date element). Temporal coverage is typically specified using named time periods (e.g., Neolithic) or the same date/time format (http://www.w3.org/TR/NOTE-datetime) as recommended for the date element.
15. *Rights:* A rights management statement, an identifier that links to a rights management statement, or an identifier that links to a service providing information about rights management for the resource.

Each Dublin Core element is defined using a set of ten attributes from the ISO/IEC 11179 standard for the description of data elements. Six of them are common to all the Dublin Core elements. A detailed description can be found in [20].

The set defines the elements for the *content* (coverage, description, type, relation, source, subject, and title) of a document. There are also elements describing the *intellectual property* rights (publisher, creator, contributor, and rights) and concrete *instantiations* (date, format, identifier, and language).

This set of elements is not sufficient for a detailed description of an information source. Here is one example: The BUSTER system offers so-called query templates that are domain-dependent to the user (e.g., land use within the geographical area). Therefore, we have to add new attributes to the basic set to refine existing attributes. Since BUSTER is based on a common vocabulary approach, a certain vocabulary has to be chosen to describe the CSD for possible future queries. This has to be included in the Dublin Core element set. In the following subsection the additional descriptions and their properties are described.

3.3.2 Additional Element Descriptions

In order to use the additional element description for further machine readable processes, we have to make sure that we use a language, which provides formal semantics (e.g., OWL [48], DAML [7], OIL [31], SHIQ [62]). We can use this kind of description logics to encode additional features such as type restrictions on slots. We use the RDF(S) syntax to ensure a wide acceptance with respect to accessibility and usability. Please note that the expressiveness of RDF(S) sometimes not enough [70]. So then, we refer to explicit ontologies available on the WWW. The following elements are refined for our CSD:

- Coverage: Since there is no further distinction between spatial and temporal coverage, this element has to be refined.
  - Spatial: The recommended best practice from DCMI [21] is to select a value from a controlled vocabulary and that, where appropriate, named places or time periods be used in preference to numeric identifiers such as sets of coordinates or date ranges. Examples are *DCMI Point* to describe a point in space using its geographic coordinates, *ISO 3166* a code for the representation of names of countries, and *DCMI Box* that identifies a region of space using its geographic limits. The last recommendation is *TGN,* the GETTY Thesaurus of Geographic Names (see http://shiva.pub.getty.edu/tgn_browser/). We also extend to possibilities introducing colloquial *place names,* which can also be structured.
  - Temporal: The recommended best practice here [21] is to use one of the two following encoding schemes: *DCMI Period,* a specification of the limits of a time interval, and *W3C-DTF,* the W3C encoding rules for dates and times - a profile based on ISO 8601 (see also: http://www.w3.org/TR/NOTE-datetime). We extend this by introducing colloquial *period names,* which can also be structured.
- Description: Description may include, but is not limited to: an abstract, table of contents, reference to a graphical representation of content or a free-text account of the content. Since this lacks formal semantics, we restrict

the description to a formal description logic, namely OWL, DAML+OIL or SHIQ. The vocabulary used to describe these A-Boxes has to be one of the vocabularies used in the "relation" element.

- Relation: The qualifiers that refine the relation element as recommended by DCMI are limited. Therefore, we need to extend these qualifiers by references that also point to ontologies, gazetteers or thesauries. A relation is described as a XML name space describing the URI of the corresponding vocabulary and a prefix to mark terms from this vocabulary.
- Subject: The qualifiers recommended by DCMI for the subject element contain common lists of keyword from various sources (e.g., the Library of Congress Subject Headings, Medical Subject Headings, Universal Decimal Classification). In BUSTER, we use the subject element accordingly, it remains a list of significant keywords to describe the information source, but the keywords have to be chosen from a controlled vocabulary referred by the relation element.
- Rights: Despite the intellectual property rights we also have to consider access rights for special user groups. In the moment, there is no further specification.

3.3.3 Background Models

In order to obtain a well defined metadata model for the information source referred to above as well as for information sources in general, we need some background models providing a standard vocabulary that can be used to describe information on a commonly agreed basis. We identify different areas of background knowledge:

- Technical background: Technical terms used in the CSD contain the additional vocabulary introduced to refine the Dublin Core standard. We refer to this additional vocabulary using the prefix **csd.** Furthermore, vocabularies for technical metadata such as type and format have to be defined. Collections of such terms exist, but they still have to be encoded formally.
- Organizational background: Information sources are always in some organizational context such as companies, administrative units or non-profit organizations. A model of the organizational context containing information about organizational units and people involved is required to describe creators, contributors and publishers of information.
- Thematic background: The most important aspect with regard to an intelligent search for information is the topic area. Terms from the specific area used to describe the content of an information source in the subject and description element have to be defined in appropriate ontologies. If concerned with complex information sources, it might be necessary to use multiple ontologies.
- Spatio-temporal background. Information often refers to a specific spatial and temporal context. In order to explicate context in an abstract

way, qualitative models of space and time are needed for references to the metadata description. In the following we use the prefix **geo** to denote the spatial and **time** to denote the temporal context.

The use of terms from the background models makes it possible to select suitable information sources on a semantic basis. For this purpose, the background models have to be made accessible to the BUSTER system as a query vocabulary.

3.3.4 Example

The example consists of an information source that can be found in the geographical area. It is a CORINE land cover database, which contains data about land use types in the southern part of Lower Saxony, Germany. We would choose a vocabulary accordingly, in this case the GEMET vocabulary [84, 29].

The header of the information source refers to the XML name spaces. The URI of the name space for the RDF schema for the RDF data model as described in the Resource Description Framework (RDF) Model and Syntax Specification is first. The URI of all DCMI elements that comprise the Dublin Core Metadata Element Set, Version 1.1 [DCMES] is listed on http://purl.org/dc/elements/1-1/. Also, we need the Dublin Core qualifiers, which are described in the resource listed below. In addition, terms we use for additional description are listed in the BUSTER CSD and the Delphi-IMM CSD. The spatial extend is given by the TGN ontology and a place name structure is given by the more general domain ontology labeled by the prefix **geo.** The geodesy ontology contains descriptions for that domain, e.g., the Bessel ellipsoid from 1841. The last name space refers to the controlled vocabulary GEMET. An example for the header is:

```
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns\#"
    xmlns:dc="http://purl.org/dc/elements/1.1/">
    xmlns:csd="http://www.semantic-translation.com/csd/ontologies/bustercsd">
    xmlns:delphi_imm="http://http://semantic-translation.com/csd/ontologies/delphi_imm">
    xmlns:tgn="http://www.semantic-translation.com/csd/ontologies/tndef">
    xmlns:geo="http://www.semantic-translation.com/csd/ontologies/ontogeodomain">
    xmlns:geodesy="http://www.semantic-translation.com/csd/ontologies/geodesydomain">
    xmlns:gemet="http://www.semantic-translation.com/csd/ontologies/gemet">
    ...
</rdf:RDF>
```

Next, we will describe all of the above mentioned elements. The elements are described accordingly in the following groups: Some elements relate to the *content* of the item, some to the item as *intellectual property,* still others to the particular *instantiation,* or version, of the item.

Content

The following N elements describe the content of the information source. The title of the information source can be described as follows:

```
<dc:title>
    Database for land use in southern Lower Saxony, Germany
</dc:title>
```

The subject is a list of significant keywords based on a known vocabulary (here: GEMET 2.0) and on the thematic background. The topics of the example would be the following:

```
<dc:subject>
    <csd:topic rdf:resource="gemet:land use"/>
    <csd:topic rdf:resource="gemet:land use classification"/>
    <csd:topic rdf:resource="gemet:landscape utilisation"/>
</dc:subject>
```

The description is the element where additional properties of the information source can be encoded. In order to keep this information machine readable and "understandable" we use OIL for the description. In the following, we add the information that this data source covers the southern part of the state of Lower Saxony, Germany with two Bessel ellipsoids:

```
<dc:description>
    <geodesy:reference rdf:resource="geodesy:Bessel-ellipsoid-1841"/>
</dc:description>
```

The type of the information source is a data set.

```
<dc:type>
    dataset
</dc:type>
```

The source refers to a second source which is given below. Please note that this has only been done to completely describe the data source.

```
<dc:source>
    http://www.tzi.de/~visser/csd/clc.dbf
</dc:source>
```

The relation element is very important. As described above, we can use background knowledge to specify the information source. In this example we need additional knowledge about the organizational structure of the company responsible for the data, some knowledge about place names defined in the geographical ontology and the formal description of some terms in the geodesy area.

```
<dc:relation>
    <csd:reference
        alias="delphi_imm"
        source=http://www.semantic-translation.com/csd/ontologies/delphi_imm.rdfs/>
    <csd:reference
        alias="geo"
        source=http://www.semantic-translation.com/csd/ontologies/geo-ontology.rdfs/>
    <csd:reference
        alias="geodesy"
        source=http://www.semantic-translation.com/csd/ontologies/geodesy-ontology.rdfs/>
</dc:relation>
```

The coverage element consist of both the spatial and the temporal description of the information source. We use the definitions given by the geographical ontology. With respect to the spatial extend we could also use the GETTY notation (if available in RDF syntax).

```
<dc:coverage>
    <geo:state rdf:resource="geo:Lower-Saxony"/>
    <geo:region rdf:resource="geo:Northwest-Germany/>
</dc:coverage>
```

Intellectual Property

In this case, the creator belongs to a private company. Since this company has its own structure, we use this structure that is listed in a CSD. The following three elements give insight in the responsibility of this source.

```
<dc:creator>
    <delphi\_imm:employee rdf:resource="delphi_imm:Ingrid.Christ"/>
</dc:creator>

<dc:contributor>
    <delphi\_imm:director rdf:resource="delphi_imm:Rolf.Lessing"/>
</dc:contributor>

<dc:publisher>
    <csd:organisation rdf:resource="delphi_imm:Delphi_IMM_GmbH"/>
</dc:publisher>
```

The rights element has an additional qualifier: the access rights of this information source.

```
<dc:rights>
    <csd:organisation rdf:resource="DelphiIMM:Delphi_IMM_GmbH"/>
    <csd:access_rights rdf:resource="csd:generally_accessible"/>
</dc:rights>
```

Instantiation

The remaining elements deal with the concrete instantiation of the information source. We use the W3C standard for the date and the Internet Media Type encoding scheme for the format. The identifier is the URI of the source and the language is encoded after ISO 639-2 as recommended by the DCMI.

```
<dc:date>
    1999-15-3
</dc:date>

<dc:format>
    application/msaccess
</dc:format>
```

```
<dc:identifier>
    http://www.tzi.de/buster/csd/clc.dbf
</dc:identifier>

<dc:language>
    en
</dc:language>
```

A few description items in the last subsection refer to additional definitions in other ontologies. These ontologies contain information about geographical names, geodesic information, organizational issues and other. In the following, we give an overview about the mentioned ontologies but focus on the important segments to cover our example, i.e., only give details of the parts of the ontology which we need to explain this example.

Delphi IMM Organization

This ontology contains the organizational structure of the private company Delphi IMM GmbH, located in Magdeburg, Germany. For better understanding the ontologies are listed in OIL-text format.

```
ontology-definitions

%
% Class-definitions
%
class-def defined Company_DelphiIMM

class-def defined Director
    slot-constraint works_for has-value Company_DelphiIMM
    slot-constraint runs has-value Company_DelphiIMM

class-def defined Employee
    slot-constraint works_for has-value Company_DelphiIMM

%
% Slot-definitions
%
slot-def runs
    properties functional

slot-def works_for
    properties functional

%
% Instances
%
instance-of Rolf.Lessing Director
instance-of Ingrid.Christ Employee

end-ontology
}
```

Geodesic Ontology

The Bessel-Ellipsoid from 1841 has three properties: the ellipticity, and the semimajor and semiminor axis. The ontology is defined in the RACER [47] language because the ontology contains concrete domains.

```
(in-knowledge-base geodesy-demo geodesy)

(signature
:atomic-concepts
    ( Ellipsoid
    )
:individuals
    ( Bessel-Ellipsoid-1841
    )
:attributes
    (
        (real ellipticity)
        (real semimajor-axis)
        (real semiminor-axis)
    )
) % end signature

%
% Class-definitions
%
(equivalent Ellipsoid
    (and (<= ellipticity    1)
         (>= semimajor-axis 6360000.0)
         (<  semiminor-axis 6360000.0)
    )
) % end equivalent

%
% Instances
%
(instance Bessel-Ellipsoid-1841
    (and (= ellipticity 0.0033428)
         (= semimajor-axis 6377397.2)
         (= semiminor-axis 6356078.9)
    )
) % end instance
```

3.4 Relevance

An important issue for the retrieval and evaluation of information source is *relevance.* Full-text retrieval algorithms are seeking strings or substrings that match the query and rank the results according to an estimate importance.

Our approach is different: at first, all the found information sources are seen as results which differ from each other through the degree of relevance with regard to the given query.

The terminological part does not contain metrics to define relevance due to the use of classifiers, which are based on description logics (cf. section 4.2). Here, we have crisp decisions: a concept is subsumed by another concept or it is not. Therefore, the calculation of the degree relevance only consists of two parts: the *spatial relevance* and the *temporal relevance*. The relevance is calculated based on a metric distance. The spatial distance consists of a formula using neighborhood and hierarchical (administrative) information. The temporal relevance is calculated from the distance of time intervals.

We define the spatial relevance in section 5.3 and the temporal relevance is discussed in section 6.3.

This page intentionally left blank

4

Terminological Representation and Reasoning, Semantic Translation

First, this section describes the requirements which we have to take into account with regard to the annotation and querying of terminological information sources. Secondly, we discuss how the terms or concepts are represented. We further propose necessary reasoning components and show how these can be used to integrate information on a conceptual level. The next subsection deals with the integration/translation on the data level, which we also refer to as context transformation. We conclude this section with an example.

This section summarizes ideas published elsewhere [132, 87]. The ontological representation and the comparison of eligible languages has been discussed in [111] and [141]. The subsection about context transformation by rules is mainly based on the work of [138]. Fundamental ideas about semantic translation has been introduced and discussed in [131].

4.1 Requirements

Both annotation and querying requirements have to be taken into account. Also, we have to bear in mind that the Semantic Web will only be successful if we can get users to annotate their data. This will only be possible if we can support them with easy-to-use tools. The following necessary requirements can be stated with regard to terminological representation (including the annotation of information sources) and reasoning.

4.1.1 Representation

As stated before, background knowledge is required in order to run "intelligent" search and/or integration engines. This knowledge has to be modeled and represented in a way that (a) the requirements of the Semantic Web are fulfilled and (b) allows us to use this representation for reasoning components.

The definition of Semantic Web reveals the most important requirement of the representation: the concepts have to have some sort of formality. This

implies that the language used in order to describe the knowledge of a domain has to have well defined semantics, which allows engines to "interpret" and process the data. The knowledge representation field offers a vast number of representation schemes that fulfill this request. Formal means that the definition of terms is written down in a formal language with well-understood semantics. Very often, a logic-based language is used for this purpose. It is important to note that the main thought behind the usage of this kind of language is the avoidance of ambiguities of concepts.

The representation schemes must also allow reasoning support. This does not necessarily imply a logic-based approach, however, logic-based approaches have advantages in this regard. One advantage is the possibility to verify the model, another advantage is the comprehensibility.

4.1.2 Reasoning

As far as reasoning components are concerned the following requirements have to be taken into account. Given a set on concepts (T-Box) in a certain structure (hierarchy) various questions should be answered. It depends on the logical semantics of the representation language what kind of questions can be answered. The following reasoning steps or inferences should be supported:

- *Consistency checking:* Is the model, i.e., the set of modeled concepts sound? Are there any contradictions in that model? This is usually done before we are able to formulate any other query. In addition to this: is there an empty set of objects described by a concept?
- *Subsumption of concepts:* This is the most important reasoning step. Are there any concepts subsumed by a given concept? In other words: does a subset relationship between a set of objects described by two concepts exist?
- *Find most general and most specific concepts:* Given a concept, the problem is to find the parents and the children of that concept. The parents are more general than the given concept and the children are more special than the given concept.

Given a set of instances (A-Box) the following inferences should be supported:

- *Consistency checking:* Are there instances with given restrictions that contradict the T-Box? An example could be value restrictions that are defined in the T-Box. If an instance of a concept of that T-Box is defined and the given value is not within the defined range, a contradiction occurs.
- *Instance individual of concept:* The question here is whether a certain object represented by an individual is a member of the set of objects represented by one certain concept of the T-Box.
- *Find concepts of instance:* Given an object represented by an individual: find the set of objects represented by concepts of the T-Box that the individual is member of.

- *Find instances for concept:* Given an object represented by a concept of the T-Box: find those objects represented by individuals of an A-Box that are member of the set of objects of the concept.

4.1.3 Integration/Translation on the Data Level

Conceptual integration of information is an important feature for future information systems running on the Web. Another important and in our opinion crucial and necessary feature is the integration/translation on the data level. We use the terms integration and translation as synonyms because we transform data from one context of information in another. Due to the fact that we do not change the original data sets but create new data sets in another context we can use the term translation. However, from an information systems point of view we could also call the process information integration.

The most important requirement is the automatic recognition of syntactic, structural, and semantic heterogeneity conflicts. Once detected, the conflict resolution part plays a crucial role. [138] differentiates between three semantic data heterogeneities:

- Unit and scale conflicts
- Representation conflicts
- Surjective projection conflicts

Unit conflicts appear if information systems contain numerical attributes with the same semantics but with different values. An example is the attribute 'size' within ATKIS-OK-25 data sets (an official landscape classification system in Germany), which contains information about the size of an area in hectares. CORINE land cover data sets (the official European counterpart to ATKIS) also contain the attribute size, however, the unit here are in acres. Scale conflicts occur if these values have different scales. An example is the attribute 'size' with the unit 1/10 of an hectare in an ATKIS-OK-25 data set and the same attribute with the same unit in an ATKIS-OK 1000 data set but with 1/1 hectares.

Representation conflicts occur if symbolic attributes also have the same semantics but differ in their values. A prominent example is the date-format in the German and British linguistic area. Usually, the German format has the form day.month.year whereas the British format is month/day/year. In order to solve this type of conflict, a conversion function has to be found that is able to convert one symbolic representation into another.

Surjective projection conflicts usually appear if we try to project elements of two different sets on each other where the number of elements in these sets differ. This means that we project one or more elements of one set onto one element of the other set. This kind of conflict is very common in practise and it is sometimes not possible to find a reasonable projection at all. An example is the subsumption process of concepts and sub-concepts in the

electronic domain. ETIM[1] and eCl@ss[2] are two official catalogue systems in Germany providing a standard for material classification in the electronic domain. The eCl@ss concepts 'Stahlpanzerrohr', 'Isolierschutzschlauch', 'Metallschutzschlauch', and 'Schutzschlauch' for example are subsumed by the ETIM 'Installationsrohr'.

It is important to detect these mentioned conflicts for the integration of information on the data level. The first two conflicts can generally be solved with a conversion function, the last conflict is harder to solve. If a projection exists, we can then find a function to solve the problem.

4.2 Representation and Reasoning Components

Knowledge representation is a wide area and cannot be considered in total. The requirements described above however suggest a logic-based approach. Therefore, we will discuss the area of ontologies and then compare some description logic dialects for the construction of ontologies.

4.2.1 Ontologies

In the early 90s a new area around the idea of ontologies began to emerge. Gruber [45] describes an ontology as a "formal and explicit specification of a conceptualization" [45, p.908]. This view of ontologies is widely accepted within the IT community. Leading researchers in the area claim that the above definition characterizes best the essence of an ontology [33]. A conceptualization refers to an abstract model of how people commonly think about a real thing in the world, e.g., a chair. Explicit specification means that the concepts and relations of the abstract model have been given explicit names and definitions. Formal means that the definition of terms is written down in a formal language with well-understood properties. Very often, a logic-based language is used for this purpose. It is important to note that the main thought behind the usage of this kind of language is the avoidance of ambiguities of concepts.

Grüninger and Uschold [43] correctly argue that there are many kinds of things that people call ontologies. Following descriptions of concepts from left to right means, we may have loose terms (less meaning) only (left part of figure 4.1). Following the line to the right the degree of meaning increases. The other extreme are descriptions of terms within formalized logical theories. Moving from left to right also means that the ambiguity of terms decreases.

The descriptions of terms in ontologies are formal as mentioned above. One can argue that the description of classes and objects represented in UML [9] is also formal. While this is true, there is a difference with regard to the degree of formalization.

[1] http://www.etim.de, verified on June 28, 2003.
[2] http://www.eclass.de, verified on June 28, 2003.

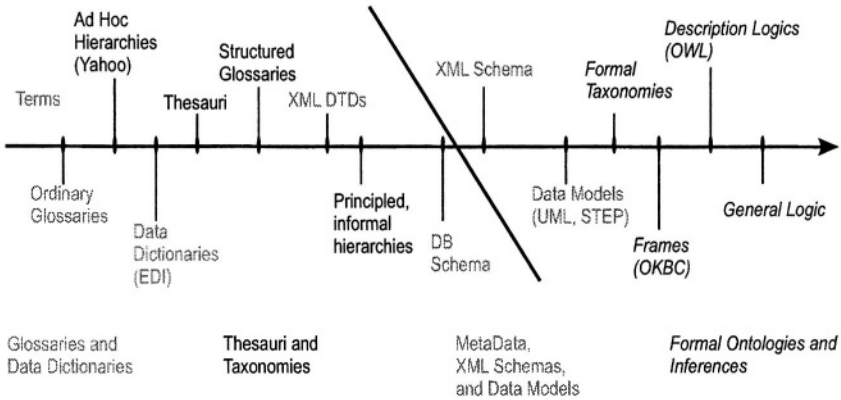


Fig. 4.1. Types of ontologies [43, p. 7, changed].

We can differentiate between informal, semi-formal, and formal languages (cf. [126]). The English language is an example of an informal language. Some terms are not well defined and it is easy to create ambiguities of concepts (e.g., "spatial boundary"). Semi-formal languages are created to support software engineers developing software systems. UML [9] is such a language but it is still open for ambiguities, whereas formal languages have a higher degree of formality. However, this does not imply that all these languages are usable for our purpose. First-order logic for example is a formal language but is undecidable in general. This language does also not contain a model-theoretical semantics, which we need for reasoning support. Most of the description logics however, support formal semantics and efficient reasoning support.

An comprehensive investigation of different approaches to intelligent information integration based on ontologies revealed the overwhelming dominance of systems using some variants of description logics as ontology representation languages [141]. We therefore compared various DL languages in regard to our requirements.

4.2.2 Description Logics

Description logics (DL) describe knowledge in terms of concepts and restrictions on roles. They can also derive classification taxonomies automatically. The main idea behind DL is to provide means to describe structured knowledge in a way that we can access and reason with it. DLs in general represent a class of logic-based knowledge representation languages. These languages represent subsets of first-order logic, which are expressive enough and also decidable in regards to inference mechanisms [85]. DLs are also known as terminological logics [4]. A specific feature of DL is that classes (concepts) can be described intensionally by properties. These properties must be fulfilled for an object to belong to this class.

Table 4.1. Expressiveness of the evaluated description logics used for information integration.

	CLASSIC	OIL	LOOM
Logical Operators			
conjunction	×	×	×
disjunction		×	×
negation		×	×
Slot-Constraints			
slot values	×		×
type restriction	×	×	×
range restriction	×	×	×
existential restriction	×	×	×
cardinalities	×	×	×
Slot-Definitions			
functional attributes	×	×	×
slot conjunction			×
transitive slots		×	
inverse slots		×	×
Axioms			
equality	×	×	×
implication	×		
disjoint	×	×	×
covering		×	
Assertions			
entities	×	(×)	×
relation-instances	×	(×)	×

Our investigation revealed that the most often cited language is CLASSIC [10], which is used by different systems including OBSERVER [81], and the work of Kashyap and Sheth [66]. Other terminological languages are GRAIL [94], (cf. the Tambis approach [106]), LOOM [78] (cf. SIMS [3]), and OIL [31].

In order to get an impression of the expressiveness of these languages, we compared them with respect to the language constructs they provide (see table 4.1 on page 58). The scope of the comparison is focused on typical constructs used in these logics. The comparison includes the use of logical operators to build class expressions, properties and constraints of slots used to describe class characteristics as well as the possibility to state terminological axioms. A further criterion is the existence of instances.

The comparison reveals an emphasis on highly expressive concept definitions. The compared languages are capable of almost all common concept forming operators. An exception is CLASSIC, which does not allow for the use of disjunction and negation in concept definitions. The reason for this shortcoming is the existence of an efficient subsumption algorithm that supports A-Box reasoning. OIL can also be used to define instances, but sound

Table 4.2. Expressiveness of the evaluated extended description logics used for information integration.

	CARIN	$\mathcal{AL}$-log	$\mathcal{DLR}$
Logical Operators			
conjunction	×	×	×
disjunction	(×)	×	
negation	×	×	×
Slot-Constraints			
slot values			
type restriction	×	×	×
range restriction			
existential restriction	(×)	×	×
cardinalities	×		(×)
Slot Definitions			
functional attributes			
slot conjunction	(×)		×
transitive slots			
inverse slots			
n-ary relations			×
Axioms			
equality			
implication			
disjoint			
covering			
Assertions			
entities		×	
relation-instances		×	
rule base	×	×	

and complete reasoning support is only provided for the T-Box part of the language. LOOM, on the other, hand provides reasoning support for A- and T-Box but it cannot guarantee soundness and completeness. Concerning the definition of slots and terminological axioms the picture is less clear.

We conclude that complex slot definitions beyond the definition of functional slots are not that important for the application at hand. Terminological axioms that seem to be important are equality and disjointness. This hypothesis can be explained by the application, where an important task is to handle synonyms and homonyms on a semantic level. We hypothesize that if the purpose is an exact definition of single terms in an information source, classical description logics do a good job in providing an expressive language and reasoning support for consistency checking and automated construction of subsumption hierarchies.

Beside the purely terminological languages mentioned above there are also approaches that use extensions of description logics that include rule bases.

Known uses of extended languages are in the PICSEL system using CARIN, a description logic extended with function-free horn clauses [40] and the DWQ project [12]. In the latter approach $\mathcal{AL} - log$ a combination of a simple description logics with Datalog is used [25]. Calvanese et al. [12] use the logic $\mathcal{DLR}$, which is a description logic with n-ary relations and is used for information integration in the same project. The integration of description logics with rule-based reasoning makes it necessary to restrict the expressive power of the terminological part of the language in order to remain decidable [72]. Table 4.2 on page 59 gives an overview on the available language constructs.

The comparison of extended description logics clearly reflects the semantic difficulties that arise from the extension. The concept definitions used are much less expressive and mainly reduced to type and existential definitions combined by logical operators. $\mathcal{AL}-log$ additionally has an A-box. Therefore, these kinds of languages can be used when the information to be represented is highly interconnected. The existence of a rule language also helps to link the ontology to the actual information.

We conclude that, if the purpose is not only to define a term, but also to capture the structure of an information source and the dependencies between information items, a rule language or n-ary relations are needed to express these dependencies.

Modern DL with efficient reasoning systems support these requirements (e.g. structure of an information source, dependencies between information items). Examples of these kinds of DL, which are also supported by an inference engine are SHIQ with the reasoner FaCT (Fast Classification of Terminologies, [59, 61] and RACER (Reasoner for A-Boxes and Concept Expressions Renamed, [47]). Both FaCT and RACER support the given requirements and are available freely. They are therefore chosen for the terminological representation.

4.2.3 Reasoning Components

The reasoning components are dependent on the logic used. However, the most important reasoning capabilities, as described in the requirements, are consistency checking, classification, and subsumption.

The FaCT reasoner is based on the SHIQ logic. SHIQ itself is based on ALC, a description logic that has been introduced by [100]. ALC allows for modeling classes and unary predicates for concepts as well as slots and binary predicates for roles. Horrocks et al. [62] extended ALC by transitive roles, role hierarchies, inverse roles, and qualified cardinalities and called this the SHIQ logic. The FaCT reasoner supports this logic and is able to support the required reasoning capabilities. Subsumption reasoning is possible on a concept level and consistency checking is also supported. The classification task, i.e., to classify whether on instance is subsumed by a concept, is supported to a certain extent. Instances are modeled as individuals but are internally handled as concepts.

RACER also supports A-Boxes explicitly. Moreover, this engine is able to handle multiple T- and A-Boxes at the same time. It is also possible to load and unload T- and A-Boxes at runtime.

4.3 Semantic Translation

The term semantic translator, a translator between information systems and/or catalogue systems (see also [130]) that gives the user the option to map data between the systems without loosing its meaning faces the problem of context dependency. Information available within a special information source can only be completely understood in the context of that information source. The conceptual view of our BUSTER approach (see figure 3.1 on page 39) allows for the use of two general methods to transform data from one context into another. First, *context transformation by rules* can be used and second *context transformation by re-classification* can be applied. We will describe these two techniques in the following sections and argue that a context transformation in practise benefits from a combination of both.

4.3.1 Context Transformation by Rules

This subsection is based on the work that has been done by Wache [137, 138]. He developed a semantic mediator called MECOTA (MEdiator with COntext TrAnsformation), which is used within the BUSTER framework (cf. figure 3.1 on page 39).

MECOTA is able to resolve structural heterogeneity conflicts. This is done with the help of rules for a reformulation of a query. Data heterogeneity conflicts can also be solved automatically using contexts (cf. [140]). The mediator uses special knowledge in the form of context transformation rules. These rules describe the conditions that have to be fulfilled in order to transform a piece of information from one context into another.

Wache introduces an integration formalism for the detection and elimination of semantic heterogeneity conflicts. This integration formalism uses a *complete* description of information. Complete, in this context, means that this description is sufficient to support context transformation.

So-called *semantic labels* are used in order to define the necessary context attributes for a certain context. A semantic label L is a semantic description of a concept and consists of a combination of the concept name and the context (e.g., units, scales). It is defined in the global vocabulary.

A *semantic label scheme* is necessary to define the extent of the context, i.e., the extent of the necessary context attributes. A semantic label scheme associates a primitive concept of the vocabulary with a context. Thus, it allows to specify, which attributes have to appear in the according semantic labels. Semantic label schemes are necessary for the completeness of a minimal context description.

Basic components of the integration formalism are so-called *templates*. Templates represent a set of complete descriptions of instances in an information system. A template consists of a name, a semantic label, a type, a value and the name of the information system (or context). A template describes not only the data but also the meaning of the data and their structure. A template is be defined as a 5 tuple:

$$M = \langle CN, L, TD, W \rangle @SN$$

where M is the template, CN is a concept name, L is a semantic label, TD is a type, W is either another template or a variable and SN is the information system [138, p. 170].

Wache [138] argues that templates are not sufficient in order to formulate queries in MECOTA. This system should also allow to restrict certain variables, i.e., use constraints. One example is to test the equivalence of two variables or to restrict a value of a variable. Thus, additional predicates $\varphi_1, \varphi_2, ...$ are used to achieve these constraints. A query (or goal) $\mathcal{G}$ is then defined as a set of goals:

$$\mathcal{G} = \leftarrow G_1, \ldots, G_n$$

where $G_i \in \mathcal{M} \cup \{\varphi_1, \varphi_2, ...\}$ and $\mathcal{M}$ is the set of templates. MECOTA accepts queries of this kind.

In general, the integration formalism is based on a rule formalism, which differentiates between two types of transformation rules:

- Rules for the reformulation of the query
- Rules for context transformation

The first type of rules follow the approach of a global scheme. This means, that a rule is generating the relation between a piece of information of a global scheme and a piece of information of a data source. The underlying principle follows the global ontology approach (see section 2.1). An original query is parsed and divided into sub-queries, which then deal with the appropriate information sources. These sub-queries are understood by the information sources and can therefore deliver the required answers. These types of queries can be used for the detection and elimination of *structural* heterogeneity problems.

Semantic *data* heterogeneity problems can be recognized and solved with the help of the second type of rules, the context transformation rules. These rules specify the knowledge needed to solve conflicts. To be more precise, a context transformation rule describes how one piece of information can be transferred from one context into another. A context consists of a set of context attributes, which describe the actual context. Having semantically equivalent contexts in different information systems implies to have the same context attributes. Otherwise, a comparison is not possible.

These two types of rules can be defined as follows [138, p. 173]: let $\mathcal{M}$ be a set of templates and $\{\varphi_1, \varphi_2, ...\}$ a set of predicates. A rule for the reformulation of a query is then defined as

$$M \leftarrow M_1, \ldots, M_n, \varphi_1, \ldots, \varphi_m,$$

and a context transformation rule has the following form:

$$M_F \rightsquigarrow M_T \leftarrow M_1, \ldots, M_n, \varphi_1, \ldots, \varphi_m$$

with $M, M_F, M_T, M_1, \ldots, M_n \in \mathcal{M}$.

The following holds for the reformulation rules: the template *M* describes that piece of information that can be generated out of all available information represented by the templates $M_1, \ldots, M_n$. In addition, the predicates $\varphi_1, \ldots, \varphi_m$ have to be fulfilled. The context transformation rule translates the template M_F in the template M_T using the additional information in form of $M_1, \ldots, M_n$ and fulfilling the predicates $\varphi_1, \ldots, \varphi_m$.

Context transformation rules are as atomic as possible to achieve a high degree of modularity. This implies that it is possible to apply more than one rule during the process of context transformation. MECOTA is able to specify the sequence of these rules automatically and is therefore highly declarative.

Wache [138] argues that the separation of the two types of rules also supports the modularity of MECOTA. In addition, there is a big chance that context transformation rules can be used in other scenarios. MECOTA connects the reformulation of the query and the context transformation in a convenient way. He argues further that, logically speaking, the reformulation of the queries into sub-queries is a resolution. The resolution is extended to a theory-resolution where the theory consists of the context transformation rules and axioms of the context transformation. To be more precise, the unification of the resolution is replaced by context transformation. This allows for the integration of conditions into the inferences of the resolution in an elegant manner.

4.3.2 Context Transformation by Re-classification

The other method for the semantic translation process is re-classification [111]. The step necessary is to explicitly represent the contextual knowledge. This can be done with the mentioned representation languages discussed in subsection 4.2.2. Transforming contexts means that we re-classify information entities using the contextual model of the target information source. This re-classification is based on the properties of an information item. We distinguish explicitly available properties directly contained as data in the information item and the properties that arise from the assumptions underlying the original information source of the entity. We derive these implicitly available properties from the contextual knowledge model provided by the information source. The property specifications are used to define necessary and sufficient conditions for concept membership.

Necessary Conditions

Classes are described by a set of necessary conditions in terms of values v_i for some properties p_i. We simply write p_i^X to denote that the specified conditions specified are fulfilled by the entity X. We claim that these properties are characteristic for that class and can therefore always be observed for instances of that class. We write $\mathcal{N}_C = \{p_1, \ldots, p_m\}$ to denote that the class c has necessary conditions $p_1, \ldots, p_m$. Assuming that class and property definitions always refer to the same entity X we get the following equation:

$$N^c \equiv c(X) \Rightarrow p_i^X, \ldots, p_m^X$$

Sufficient Conditions

On the other hand, we assume that an entity automatically belongs to the class c if it shows sufficient characteristic properties. We write $\mathcal{S}_C = \{p_1, \ldots, p_n\}$ to denote that $p_1, \ldots, p_n$ are sufficient conditions indicating that X belongs to the concept C. We characterize the class c by the following equation:

$$S^c \equiv p_1^X, \ldots, p_n^X \Rightarrow c(X)$$

The distinction between necessary and sufficient conditions for concept membership enables us to identify entities that definitely belong to a concept because they show all sufficient conditions. Reversely, we can identify entities that clearly do not belong to the concept, because they do not fulfill the necessary conditions.

Classes identify common properties of their members by defining necessary conditions for a membership. A classification problem is characterized by the determination of membership relations between an object and a set of predefined classes. The identification process starts with data about the object that has to be classified. This data is provided by so-called observation. In the course of classification, the observed data are matched against the necessary conditions provided by the class definitions leading to one or more classes. The match between observations and membership conditions is performed using knowledge that associates properties of objects with their class.

Stefik [105] formalized this classification view in the following way : C is a set of solution classes, O is a set of observations, and R is a set of classification rules.

In our case, the solution classes are the concept predicates $\{c_1, \ldots, c_m\}$. The observations are the necessary conditions for concept membership $\{N^c | c \in C\}$ that we derive from the specification of the contextual knowledge of the source and the properties of an entity directly encoded in the information source. The classification rules are the sufficient conditions for class membership $\{S^c | c \in C\}$ specified in the contextual knowledge of the target information source that we use in order to decide whether an entity belongs to a certain concept.

In principle a classification task is then to find a solution class $c_i \in C$ in such a way, that

$$O \cap R \Rightarrow c_i(X)$$

These given definitions cause that semantic translation is equivalent to a re-classification of entities that are already classified in one semantic structure $S = \{c_1^S, \ldots, c_n^S\}$ using another semantic structure $T = \{c_1^T, \ldots, c_m^T\}$. The process of re-classification can be based upon the semantic characterizations given by both structures. While the definitions in the source structure S can be used to infer properties of an entity, the semantic characterizations of concepts in the target structure T define the goal that has to be proven to classify an entity into an existing concept in T.

These two methods are related to the general BUSTER approach as seen in figure 3.1 on p. 39 in the following way: a mediator-wrapper architecture, which has been developed and implemented within the MECOTA project is used to transform the actual data. However, the necessary context transformation rules are generated on the semantical level by the mapper and the CTR engine. The mapper re-classifies the concepts from one context into another and the CTR engine writes the context transformation rules that are then used by the mediator MECOTA.

In summary, context transformation rules are useful for functional transformation, e.g., conversion functions. This approach can also be used for small sets of data. The main reason for this is that the context transformation rules have to be written manually. If large sets of data have to be transformed, e.g., for catalogue integration, the re-classification approach is useful. As we have seen, we classify a context description into a goal structure. We use a description logics classifier to realize this task.

4.4 Example: Translation ATKIS-CORINE Land Cover

This example covers the semantic translation service, which is available on the result panel (cf. section 7) after formulating the query and finding eligible data sets. Note: this example does not cover the conceptual integration within the search process. However, the overall scenario is described for better understanding. This example is based on the following assumptions:

- The information sources are annotated according to the comprehensive source description described in section 3.3.
- Ontologies are available for the two catalogues systems described below.

Our example covers a scenario with real-world data facing the problem of catalogue integration within a geographical domain (see also [87]. Geographical information systems normally distinguish different types of spatial objects.

Standards (also called catalogues) exist that specify these object types. Since there is more than one standard, these catalogues compete with each other. To date, no satisfactory solution has been found to integrate these catalogues.

We use two catalogue systems, namely the German ATKIS-OK-1000 [1] and the European CORINE (Co-ordination of Information on the Environment) land cover catalogue [30]. The ATKIS catalogue is an official information system in Germany. It is a project of the surveying offices of all the German states. The working group offers digital landscape models with different scales from 1:25.000 up to 1:1.000.000 with a detailed documentation in corresponding object catalogues. We use the large scale catalogue OK-1000. This catalogue offers several types of objects including definitions of different types of areas. The left part of figure 4.2 shows the different types of areas defined in the catalogue.

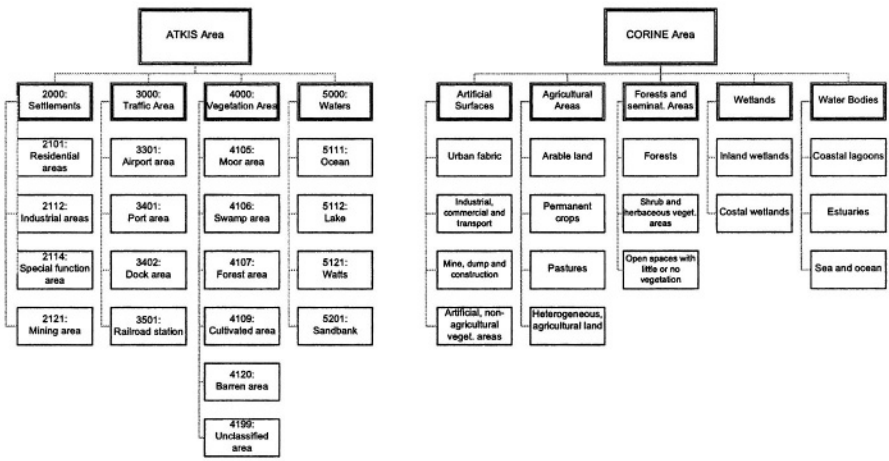


Fig. 4.2. Taxonomy of land-use types in the ATKIS-OK 1000 and the CORINE land cover catalogues.

CORINE land cover is a deliverable of the CORINE Programme the European Commission carried out from 1985 to 1990. The results are essentially of three types that correspond to the three aims of the programme:

- An information system on the state of the environment in the European Community has been created (the CORINE system). It is composed of a series of data bases describing the environment in the European Community, as well as of data bases with background information.
- Nomenclatures and methodologies were developed for carrying out the programs, which are now used as the reference in the areas concerned at the Community level.

- A systematic effort was made to concert activities with all the bodies involved in the production of environmental information especially at international level. The nomenclature developed in the CORINE programme can be seen as another catalogue, because it also defines a taxonomy of area types (see figure 4.2 on the right)

The taxonomies of land-use types in figure 4.2 illustrate the context problem mentioned earlier. The set of land types chosen for these catalogues are biased by their intended use: while the ATKIS catalogue is used to administrate human activities and their impact on land use in terms of buildings and other installations, the focus of the CORINE catalogues is on the state of the environment in terms of vegetation forms. Consequently, the ATKIS catalogue contains fine-grained distinctions between different types of areas used for human activities (i.e., different types of areas used for traffic and transportation) while natural areas are only distinguished very roughly. The CORINE taxonomy, however, contains many different kinds of natural areas (i.e., different types of cultivated areas) which are not further distinguished in the ATKIS catalogue. On the other hand, areas used for commerce and traffic are summarized in one type.

A possible scenario involves a typical user, a staff member working in an environmental department of a local authority. One of the data sets they are using concerns the object types of the German landscapes. These kind of data are usually updated and classified after the ATKIS catalogue system. The user maintains the data set with a GIS (ArcView in our case). The area involves the small town of Bad Nenndorf south-west of Hannover in Lower Saxony. Figure 4.3 shows the object type classification of the landscape in and around this town[3]. Please note also that the covered area is about 5 × 6 km big.

A functional view on the BUSTER system might help to understand the processes involved in this scenario. Figure 4.4 gives an overview. At the top, any application can incorporate or call the BUSTER client. Here, we deal with a GIS. The bottom of figure 4.4 shows both the search (on the left) and translation part (on the right). The BUSTER client will be able to connect to the BUSTER system over the Internet and activate one of the services provided.

Suppose the user is seeking some new information about the area they are interested in (Bad Nenndorf in this case). The ATKIS data sets are usually updated every 5-10 years, hence, there is a good chance of having fairly old data. However possibly, there might be actual satellite images containing landscape information that cover the area. The user is asked to restrict the defining properties of a data source in order to restrict the set of all infor-

[3] Our example area is actually classified after ATKIS-OK 25, which is a larger scale than the CORINE data sets. We will see the differences when comparing the data in figure 4.7 on page 73. The main reason for this minor problem is that the used data sets were free of charge. However, this does not influence the demonstration of our approach.

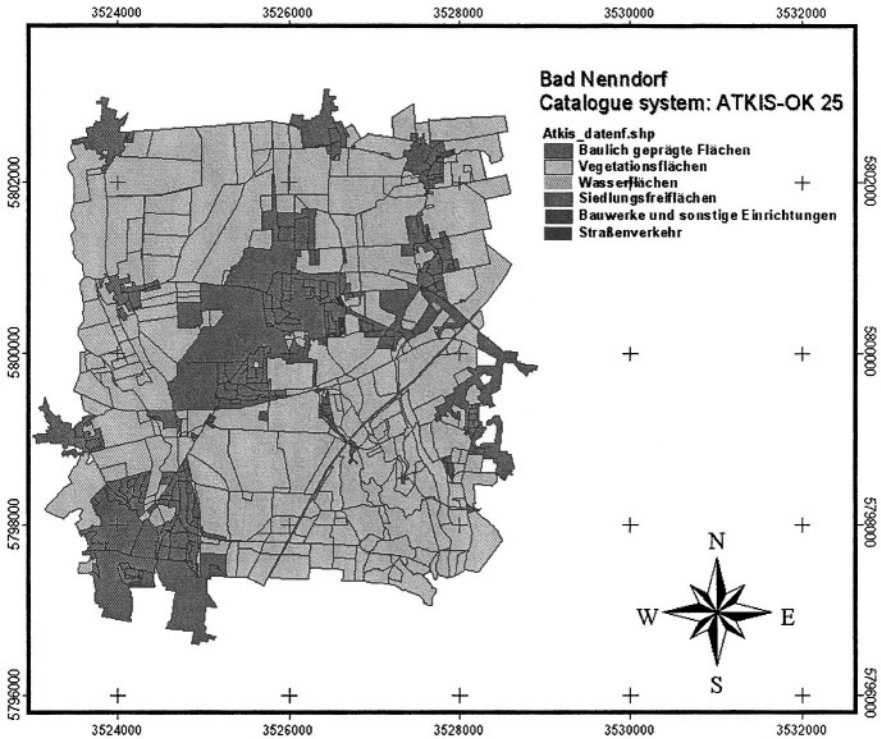


Fig. 4.3. ATKIS data set of Bad Nenndorf, a small town in the south-west of Hannover, Germany.

mation sources to those of interest. Now, the user queries BUSTER. In our example, the FaCT reasoner is the main inference engine of the BUSTER system. The resulting class definition is passed to the reasoner which places the query in a hierarchy of classes. Each class is a surrogate for an information source. All classes placed in the subtree rooted at the query class are returned, because they fulfil the constraints defined in the query. The BUSTER system presents the information sources matching the query.

The information source is labeled and several services are shown. In this case, the user can now either directly view the information as an image, or define a target file format the information source should be converted to. Currently, for displaying an image FME [98] is used to create the output format. For the semantic transformation any configured mediator could be used - as a standard we use MECOTA.

The available services highly depend on the description of the particular information source. In our example, the system offers a CORINE-To-ATKIS translation because it already 'knows' that the employee is dealing with ATKIS data and it has found new data classified after CORINE land

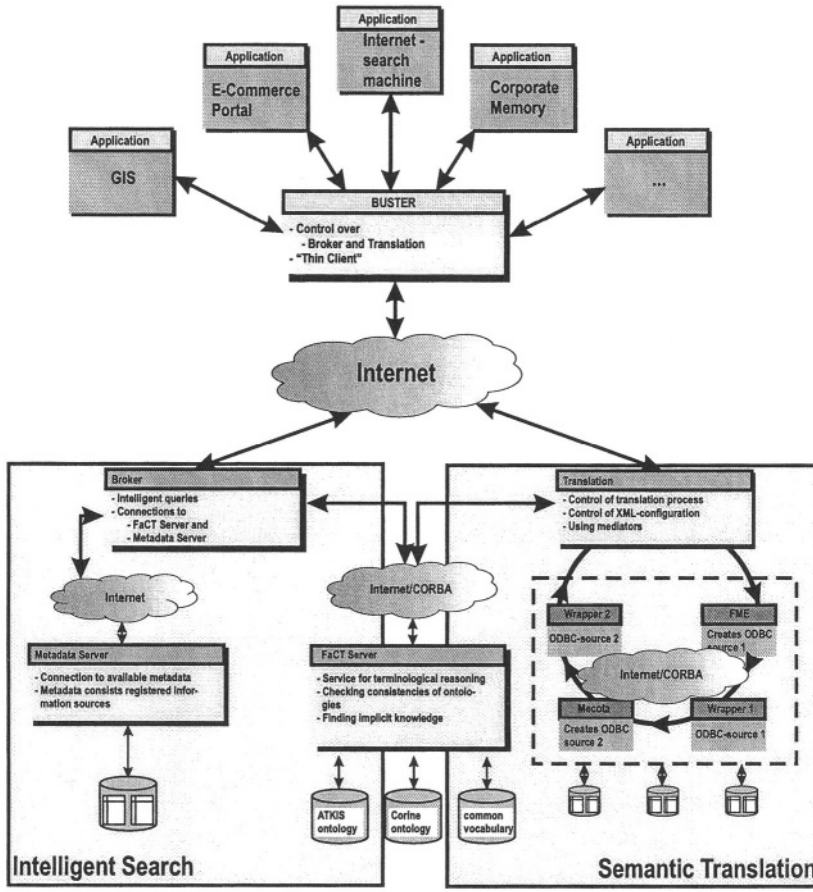


Fig. 4.4. Functional view of the BUSTER system (for the described example).

cover. CORINE land cover is a landscape classification scheme that has been defined by the European Environmental Agency [30]. Figure 4.5 shows the CORINE land cover scene, which is a classified satellite image of the southern part of Lower Saxony containing landscape information.

In our example the area covered from the information source found is way to big. As a comparison: the Bad Nenndorf area that is covered by the ATKIS data as seen in figure 4.3 is about 5 × 7km. The satellite image on the other hand covers about 91 × 45km. Therefore, we need another mediator, which is able to select the appropriate area. We use the Feature manipulation Engine (FME) for this purpose[4]. This mediator writes the data to a temporary file.

[4] Safe Software Inc. itself offers semantic data translation with their product Feature Manipulation Engine [97]. However, their white papers about semantic data translation reveal that the translation process is done manually.

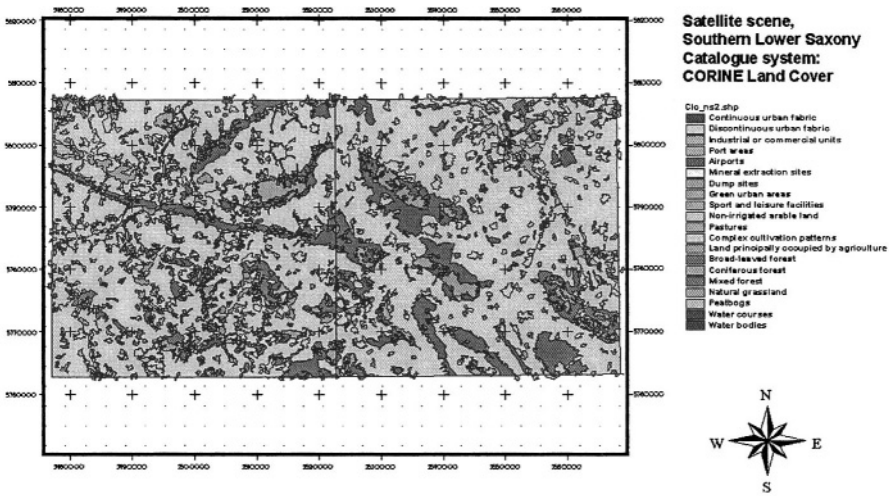


Fig. 4.5. Satellite image of the southern part of Lower Saxony containing landscape information, classification scheme is CORINE Land cover.

BUSTER offers a semantic translation because the CSD (see also 3.3) provides a link to the ontologies. An example is the description of the class *Mixed-Forest* in the CORINE catalogue (313). Mixed-Forest describes an area, which is covered with vegetation. The vegetation consists of forest trees and is cultivated. In terms of our re-classification approach the sufficient conditions are coded as follows: $S_{Mixed-Forest} \equiv has\_vegetation(X, Forest-Trees) \wedge is\_cultivated(X, true)$ for an area *X*. *Has_vegetation* and *is_cultivated* are 2-ary predicates from the common vocabulary. Their arguments *Forest – Trees* and *true* are also in this vocabulary.

The ATKIS-OK 25 ontology consists of a class *Forest,* which has the necessary conditions that the area is either covered with forest plants or has the vegetation grass with the grass being cultivated. Both cases also include the size of that area, which has to be at least 10 hectares. This can be coded as follows:

$$N_{Forest_1} \equiv has\_vegetation(X, Forest-Plants) \wedge size(X, ?SIZE) \wedge ?SIZE > 10$$

$$N_{Forest_2} \equiv has\_vegetation(X, Grass) \wedge is\_cultivated(X, true) \wedge size(X, ?SIZE) \wedge ?SIZE > 10.$$

A CORINE land cover area that is classified with 313 (for Mixed-Forest) and is greater than 10 hectares can be re-classified to the ATKIS-OK 25 class 4107 (for Forest-Area) because *Forest – Plants* are a super-concept of *Forest – Trees* (cf figure 4.6) because the size of 25 hectares is greater than the necessary condition 10.

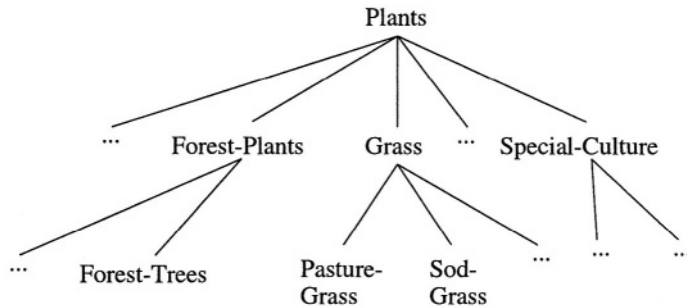


Fig. 4.6. Part of the vegetation ontology.

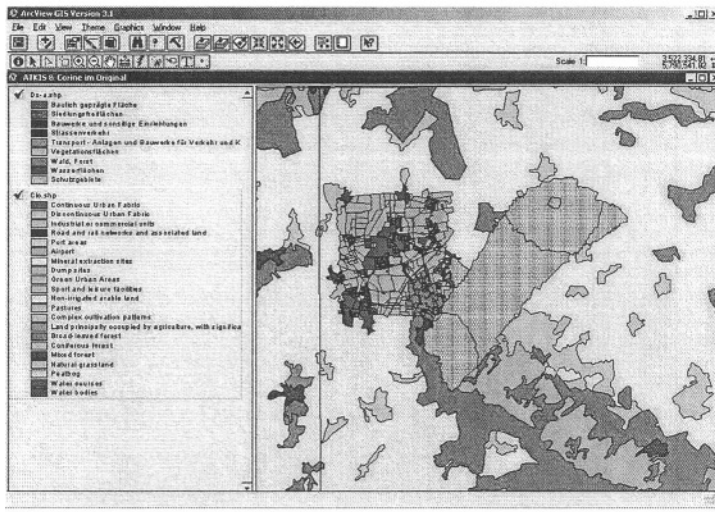
The result of the mapping task is a complete re-classification of those concepts found in the source context that can be proven right for the target concept. We have chosen the ATKIS-OK 1000 catalogue systems as the target context in order to compare the ATKIS and CORINE data appropriately.

The next step is to transform the data from the source to the target context. This is done with context transformation rules. The mapper creates a PROLOG file, which contains the re-classification results. This file implements the 'classify'-predicate in MECOTA, which is used to transform the data. For our concrete example the file contains the following lines, please note that our ATKIS area is small and our CORINE land cover data therefore only contain four ATKIS categories:

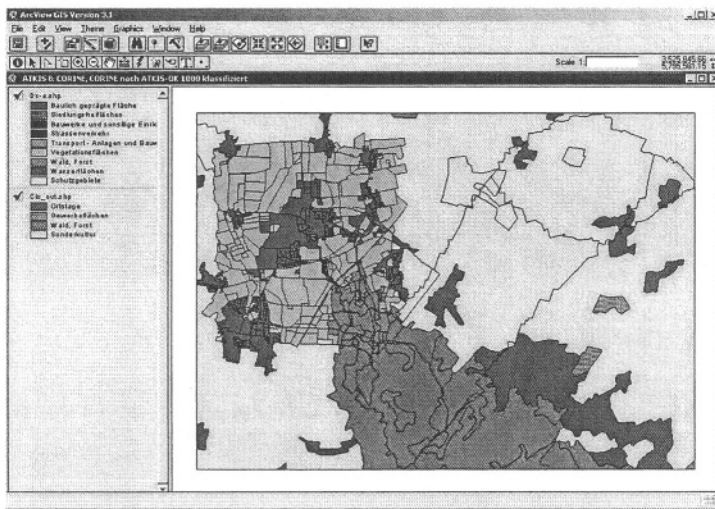
```
% CORINE: Discontinuous urban fabric (211)
% ATKIS: Ortslage (2101)
classify(VAR_CORINE_VALUE,
     _VAR_CORINE_LABEL,
     VAR_ATKIS_VALUE) :-
    VAR_CORINE_VALUE = 112,
    !,
    VAR_ATKIS_VALUE = 2101.

% CORINE: Mineral extraction sites (131)
% ATKIS: Bergbaubetrieb (2121)
classify(VAR_CORINE_VALUE,
    _VAR_CORINE_LABEL,
    VAR_ATKIS_VALUE):-
```

```
    VAR_CORINE_VALUE = 131,
    !,
    VAR_ATKIS_VALUE = 2121.

% CORINE: Broad-leaved forest (311)
% ATKIS: Wald, Forst (4107)
classify(VAR_CORINE_VALUE,
     _VAR_CORINE_LABEL,
     VAR_ATKIS_VALUE) :-
    VAR_CORINE_VALUE = 311,
    !,
    VAR_ATKIS_VALUE = 4107.

% CORINE: Mixed forest (313)
% ATKIS: Wald, Forst (4107)
classify(VAR_CORINE_VALUE,
     _VAR_CORINE_LABEL,
     VAR_ATKIS_VALUE) :-
    VAR_CORINE_VALUE = 313,
    !,
    VAR_ATKIS_VALUE = 4107.

% CORINE: Non-irrigated arable land (211)
% ATKIS: Sonderkultur (4109)
classify(VAR_CORINE_VALUE,
     _VAR_CORINE_LABEL,
     VAR_ATKIS_VALUE) :-
    VAR_CORINE_VALUE = 211,
    !,
    VAR_ATKIS_VALUE = 4109.
```

MECOTA calls a wrapper to get the input file, and a second wrapper to write the output file in the desired format. In between, the context transformation is performed. The data created will now be saved on the user's computer and the process of brokering and retrieving of the desired information is completed. The user is now able to work with the new data. Figure 4.7a shows the data before the context transformation, after running FME. We see the ATKIS map laying on top of the CORINE map. Figure 4.7b shows the final result. The CORINE data have been transformed into the ATKIS context.



(a) Before context transformation



(b) After context transformation

Fig. 4.7. The original ATKIS data with the found CORINE land cover data before and after context transformation to the ATKIS classification scheme.

This page intentionally left blank

5

Spatial Representation and Reasoning

This section describes the requirements which we have to take into account with regard to the annotation and querying of spatial information sources. Following, we discuss how our qualitative abstraction of space is represented. Spatial relevance is an important feature concerning neighborhood and partonomic distance. This is discussed in the following subsection and demonstrate the performance of this approach with examples.

This chapter summarizes ideas that have been published elsewhere [113, 128, 121, 133, 120]. The first ideas and the polygonal representation have been introduced in [112, 99]. The fundamental ideas behind the components and the functionality of the spatial reasoner have also been extensively discussed with Thomas Vögele and Christoph Schlieder.

5.1 Requirements

Annotation and retrieval of spatial information should be more flexible, comfortable, and improve situations in practise. Both the knowledge engineer and the user should have several options to annotate or retrieve information for their use. Following, we describe the necessary requirements and discuss them in short.

5.1.1 Intuitive Spatial Labeling

With intuitive labeling, the most important requirement is the option to label spatial objects/regions with intuitive names. These names should be published and can therefore act as reference intervals for further internal or external use. The majority of data available on the Web has a reference to a geographical region to some extent. We can distinguish between *directly* and *indirectly* geo-referenced information.

Direct geo-referenced information can be found as digital maps in geographical information systems. Sensor data that are used for location-based

services or robot navigation is usually directly geo-referenced. Another important way of geo-referencing data is to use indirect information. This means that the data is geo-referenced by place names, i.e., common names for locations, areas or regions without a reference system in the background. An example for this is the area of "Das Viertel" in Bremen: it would be hard to find somebody in Bremen who does not know where the area is, however, there is no such area listed on any official map of Bremen. These kinds of indirect geo-referenced information is used by digital libraries (e.g., Maryland Digital Library, Alexandria Digital Library). Also, the Semantic Web also requires techniques able to process this kind of geographical information.

The challenge is to provide integrated access to both directly and indirectly geo-referenced information.

5.1.2 Place Names, Gazetteers and Footprints

Existing GIS require direct geo-referenced information in order to implement the users queries. However, this does not fulfil the requirements necessary to process information on and for the Semantic Web. Therefore, we would like to offer users some kind of natural language identifiers for geographic objects. These are known as common geographical names or *place names* and can be seen as instances of geographic concepts. We distinguish between different types of place names:

- *Standardized place names:* these are usually long-term place names for geographical objects such as "Bodensee (Lake Constance)". These names almost never change.
- *Colloquial place names:* these are valid for a limited period within a local user community, examples are "Das Viertel" in Bremen or "Quartier Latin" in Paris.
- *"Ad-hoc" place names:* these are short-term place names valid within a limited user community. Examples are "The Deep South" (Smithonian Guides to Historic America, [76]) or "our neighborhood".
- *Activity-induced place names:* this kind of place names are usually spatio-thematic regions such as "Exhibition Hall 1" or "Exhibit A" within museums.

Place names are a user-friendly and, from a cognitive perspective, sound method to both annotate spatial metadata and specify spatial queries. In general, place names are organized in place name lists, or gazetteers.

Gazetteers use spatial footprints of reference place names to geographic locations [57, 96]. Typical examples of widely used gazetteers are the Getty Thesaurus of Geographic Names[1], the GEIN gazetteer[2], or the Alexandria Digital

[1] http://www.getty.edu/research/tools/vocabulary/tgn/, verified on May 10, 2003.
[2] German Environmental Information Network, http://www.gein.de, verified on May 10, 2003.

Library Gazetteer (ADL)[3]. Recently, standardization efforts from different organizations with regard to gazetteers have appeared. Hill [56] describes the ADL Gazetteer Content Standard, the ISO discusses the ISO/DIS 19112 Geographic information - Spatial referencing by geographic identifiers)[4], and the Open GIS Consortium also develop a standard, the OGC Web Gazetteer Service (WGS)[5]. Components of a gazetteer are a *name,* a *type* (concept), and a *footprint* (location).

Because of the simple spatial encoding they use, most state-of-the-art gazetteers possess only poor spatial reasoning capabilities. To overcome this limitation, we need to develop a new representation scheme for place names in the form of *place name structures* based on qualitative spatial models [99,119]. In addition, we need a new type of spatial footprint in order to achieve our goals.

5.1.3 Place Name Structures

Place name structures (PNS) are tools to organize and manage place names. They form a place name partonomy, which can be seen as a representation of a conceptual view. Some place names in the PNS are *extensionally* defined as an approximation of a location within a reference tessellation (footprint). However, we would also like to be able to determine *intentional* relations between place names in a PNS. From the modeling point of view, PNS are intuitive since humans tend to think in structures rather than lists.

PNS are an extension of gazetteers and should not be mixed up with "ontologies of geographic kinds", which are discussed in [104]. PNS are a specific conceptualization of a geographical space, but on an instance rather than on a conceptual level.

5.1.4 Spatial Relevance

Tobler stated in his 'first law of geography' that *"everything is related to everything else, but near things are more related than distant things"*. The distance between points or regions in space is important, however, we would like to focus on the term "proximity" which describes the distance on a more qualitative level.

Being on a qualitative level allows us to integrate other issues in order to define this term. An example of this is 'hierarchical information' such as administrative units. Combining this partonomic information with the topological (or neighborhood) information enables us to define more sophisticated queries and leads to a new type of spatial relevance.

[3] http://fat-albert.alexandria.ucsb.edu:8827/gazetteer, verified on May 10, 2003.

[4] Draft Version, http://www.isotc211.org/publications.htm, verified on May 10, 2003.

[5] OGC Gazetteer Service Draft Candidate, http://www.opengis.org/techno/discussions/02-076r3.rtf, verified on May 10, 2003.

Adding partonomic information in order to determine the spatial relevance is also supported from a cognitive perspective. [58] argue that humans arrange information in hierarchies and use hierarchical methods for their reasoning. In addition, geographical objects (both fiat and bona fide) are typically organized in partonomies as described in [116].

This type of spatial relevance should allow overriding topological relevance when weighting the partonomic part accordingly, however, this is highly dependent on the context of the query.

5.1.5 Reasoning Components

Spatial inference makes implicit spatial information explicit. This does not necessarily require a logical framework. In fact, computational approaches for spatial reasoning often take a different approach to inference, for example constraint satisfaction (cf. [16] for an overview of this line of research).

A basic inference problem derives from the standard way that gazetteers are used. For instance, a query is formulated which contains a place name. The user expects the system to return a ranked list of footprints, which contain references to information items relevant to the place name in the query. One footprint, obviously, to return is the polygon which is designated by the name. Other footprints are those close to this polygon in respect to the metric in the graph. The elementary inference step consists of determining the information about the relative position of the polygons, which are the neighbors, and which are the neighbors of their neighbors etc. of a polygon. This type of inference deals with topological information. Frequently, topological reasoning is formalized by representing the problem in the relational algebra of the region connection calculus (RCC) [93, 26]. A finite domain constraint solver is used to compute the inferences. This however is reasoning about topology. Our new type of spatial relevance therefore requires a new type of reasoning: *spatial relevance reasoning.*

5.2 Representation

As we have seen, all standard footprint representations have serious shortcomings. The solution we propose to this problem consist of introducing a new type of footprint that allows us to combine the advantages of exact polygon representations with the efficiency and ease of use of the less exact representations. This new footprint is based on a qualitative representation of polygon data.

5.2.1 Polygonal Tessellation

More sophisticated gazetteers need some information about the spatial arrangement of the footprints and therefore can not rely on bounding boxes. In

order to find a type of abstraction adapted for the use of polygonal footprints, we examine at different geometrical arrangement of polygons. In the following, polygons are closed sets of points, i.e., edges and vertices belong to the polygon.

We consider polygons $P_1, \ldots, P_n$ that are contained in a part of the plane bounded by a polygon P. Two special types of arrangements[6] of the polygons within the containing polygon P can be distinguished:

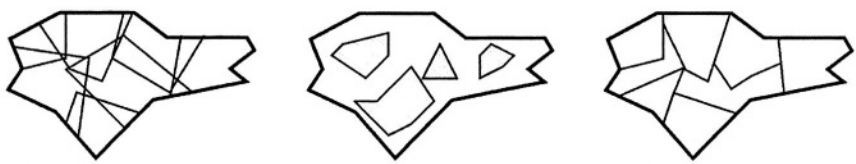
Definition 5.1
In a polygonal covering $P_1 \cup \ldots \cup P_n = P$, *the polygons* $P_1 \cup \ldots \cup P_n$ *cover the containing polygon* P.

In general, they will overlap.

Definition 5.2
A polygonal patchwork $interior(P_i \cap P_j) = \emptyset$ *for all* $i \neq j \in \{1, \ldots, n\}$. *The polygons are either disjoint or intersect only in edges and/or vertices.*

Definition 5.3
A polygonal tessellation *is a polygonal covering which also forms a polygonal patchwork.*

Polygonal tessellations arise frequently in connection with geographic footprints of place names: in a map of Germany, for instance, the federal states constitute a tessellation. Because of their importance to gazetteers, we will pay more attention to tessellations than to any other arrangement of spatial parts. Figure 5.1 shows the three polygonal arrangements.



(a) Polygonal covering: $P = P_1 \cup \ldots \cup P_n$

(b) Polygonal interior: $P > P_1 \cup \ldots \cup P_n, P_i \cap P_j = 0$ for all $i \neq j \in \{1, \ldots, n\}$

(c) Polygonal tessellation

Fig. 5.1. Polygonal arrangements in a plane.

If Π denotes the set of polygons in the plane, then a binary relation *part-of* $\subseteq \Pi \times \Pi$ can encode the fact that a polygon is a spatial part of another one, but not the fact that the polygon together with others constitutes a covering, patchwork, or tessellation. To capture these distinctions, we define relations *covering, patchwork,* and *tessellation.*

[6] Note: Polygons can be arranged in P so that they neither form a covering nor a patchwork.

Definition 5.4
The relation $covering(P, P_1, \ldots, P_n) \subseteq \Pi \times 2^{\Pi}$ *holds true iff* $\{P_1, \ldots, P_n\}$ *is a covering of* P.

Definition 5.5
The relation $patchwork(P, \{P_1, \ldots, P_n\}) \subseteq \Pi \times 2^{\Pi}$ *holds true iff* $\{P_1, \ldots, P_n\}$ *is a patchwork of* P.

Definition 5.6
The relation $tessellation(P, \{P_1, \ldots, P_n\}) \subseteq \Pi \times 2^{\Pi}$ *holds true iff* $\{P_1, \ldots, P_n\}$ *is a patchwork of* P.

Partonomies are the result of recursively applying standard part-of relation to describe parts of parts. Similarly, the polygons of a covering, patchwork or tessellation can contain other polygons. In analogy to partonomies we introduce decompositions which are defined recursively as hierarchical data structures for encoding the spatial part-of relation together with the type of arrangement of the parts.

Definition 5.7
Let $\mathcal{D} = \{D_1, \ldots, D_k\}$ *be a set of decompositions,* $\mathcal{P}$ *a polygon, and* r *a relation symbol from* $r \in \{undecomposed, tesselation, patchwork, covering\}$. *A triple* $(P, r, \mathcal{D})$ *then is a decomposition of the polygon* P *where all* $D_i = (P_i, r_i, \mathcal{D}_i)$ *satisfy one of the following conditions:* $r = undecomposed$ *and* $\mathcal{D} = \emptyset$; ...; $r = covering$ *and* $covering(P, \{P_1 \ldots P_k\})$. *A decomposition is called* homogeneous *iff it consists of a single type, that is, only one kind of relation symbol is used.*

By abstraction from this type of spatial arrangement, one obtains the partonomy that underlies a decomposition. This partonomy is encoded by the *decomposition tree* which has the same nodes as the decomposition and whose edges denote the binary part-of relation between polygons (fig. 5.2).

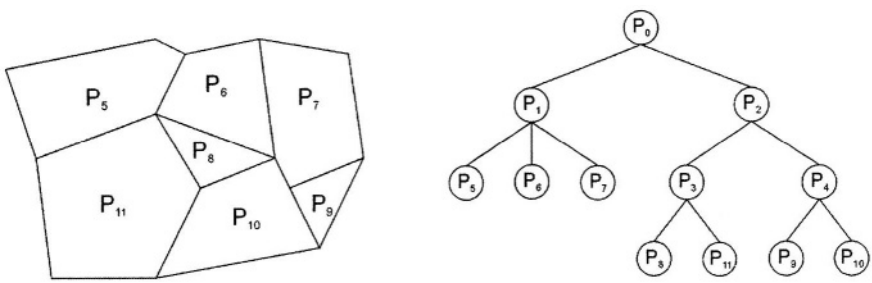


Fig. 5.2. Homogeneous decomposition by tessellation.

Vögele and Schlieder[119] defined the relation $tess \subseteq \Pi \times 2^{\Pi}$, where Π denotes the set of polygons in the plane, and $tess(P, \{P_1, \ldots, P_n\})$ *iff*

$\{P_1, \ldots, P_n\}$ is a tessellation of P. This definition allows us to say that a polygon P_1 is *part-of (po)* a polygon P_2 if P_1 is part of the decomposition by tessellation of P_2. An example might clarify what this means: figure 5.3 shows a tessellation with the decomposition hierarchy. We can say that $AA\ po\ A$, and $tess(A, \{AA, AB\})$. Another example is $CBApoCB$, $CBBpoCB$, and $tess(CB, \{CBA, CBB\})$.

This kind of representation has several advantages. These *polygonal standard reference tessellations* (pSRT) can be found easily in the real world. They are mostly artificial, man made and form organizational hierarchies. Examples are administrative units such as countries, states, and counties. Postal codes, telephone area codes are also pSRTs. These decompositions are usually well-know, intuitive, and, important if not necessary, digitally available. pSRTs are also the standard format in geographical information systems. This allows us to use the polygons that already exist in GIS. Please note that this is a big advantage considering the fact that we would like to use existing data with regard to the Semantic Web. In addition, we are able to automatically generate these polygons into a graph representation needed for our approach. Lastly, using a graph representation approach gives us the opportunity to use already existing graph algorithms such as Dijkstra's shortest path algorithm [24].

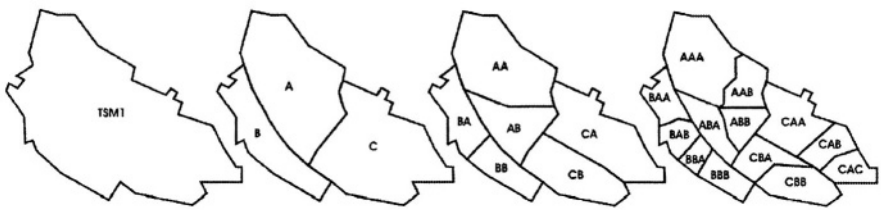


Fig. 5.3. Tessellation and decomposition hierarchy.

5.2.2 Place Names

A central idea behind gazetteers is that they give the user access to information items based not just on thematic but also on spatial relevance. This raises the computational problem of deciding which geographic footprints are relevant in respect to a given footprint. Generally, the problem is solved by defining an appropriate metric on the space of geographic footprints. For points, a chessboard metric is easily obtained by superimposing a grid onto the map space. Points lying in the same grid cell as the given point (distance 0) are considered most relevant; next come points from the four immediately neighboring cells (distance 1).

We concentrate on the most important case, homogeneous decomposition by tessellations.

Footprints

Footprints are essential for spatial reasoning capabilities of gazetteers. Most state-of-the-practice gazetteers, however, use simple types of footprints. We distinguish between (a) point, (b) bounding box, and (c) polygons. They are shown in figure 5.4. All footprints use geographic coordinates. They are either complex and require a high data volume with high computational costs or are rather simple with limited spatial reasoning capabilities. Some state-of-the-

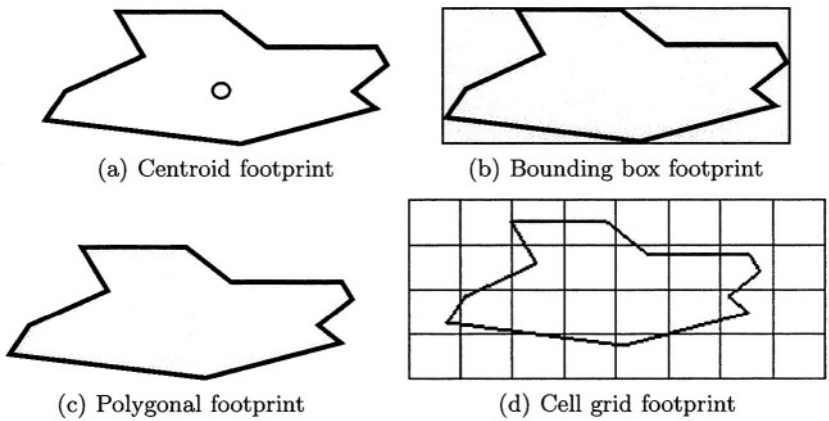


Fig. 5.4. Various footprints using geographical coordinates and spatial indices.

art gazetteers (e.g., GEIN[7]) use other types of footprints using spatial indices based on a *uniform reference grid.* Figure 5.4d shows an example. A clear advantage of these kinds of footprints is the ability to represent undeterminate boundaries. Also, topological relations as well as distances can be inferred. On the other hand, their fixed grid dimensions are rather counter-intuitive and from the GIS point of view they are no standard.

These shortcomings lead to a new type of footprint based on a pSRT as defined in section 5.2.1. Figure 5.5 gives an example. In the following, this footprint is used. The polygons in the plane will also be referred to as *reference units.* Place names are *extensionalized* in terms of reference units which simply means that there exists a binary relation between a place name and a reference unit.

In a homogeneous decomposition by tessellation two kinds of structure with spatial character interact. Firstly, there is the recursive structure of the decomposition reflected by the decomposition tree. Secondly, there exists a neighborhood structure due to fact that a polygon shares each of its edges or each of its vertices with at most one other polygon.

[7] http://www.gein.de, German Environmental Information Network, verified on June, 1st, 2003.

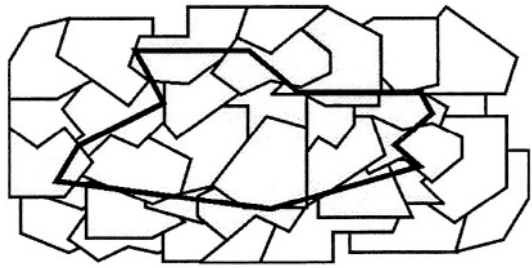


Fig. 5.5. Reference tessellation footprint.

Neighborhood Graph

We focus on direct neighborhoods, i.e., neighbors defined by shared edges. The following example shows that P_6, but not P_5, is a neighbor of P_8 because these polygons possess a common edge. The neighborhood structure is expressed by a graph (fig. 5.6).

Definition 5.8
The neighborhood graph of a homogeneous decomposition by tessellation is a graph $\mathcal{G}_{\mathcal{N}} = (V_{\mathcal{N}}, E_{\mathcal{N}})$ with the set of undecomposed polygons as nodes $V_{\mathcal{N}}$ and all pairs of neighboring polygons as edges $E_{\mathcal{N}}$.

If there is no interesting information items linked to a polygonal footprint, a good place to search for further information are its neighboring polygons. Alternatively, one could search in those polygons that are part of the same decomposition. Obviously, this leads to two different criteria of spatial relevance, which will be discuss later. In other words, a spatial relevance metric can be based on either the decomposition tree or the neighborhood graph (fig. 5.6). [99] discussed the issues about inferring relevance from spatial neighborhood and concluded that known approaches based on neighborhood graphs such as the RCC calculus are not sufficient enough to provide satisfactory results if using planar polygons as a basic model.

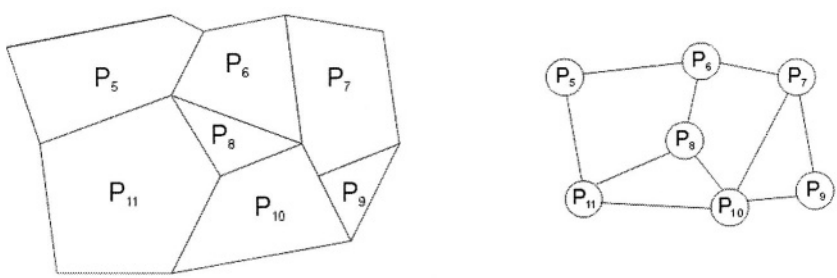


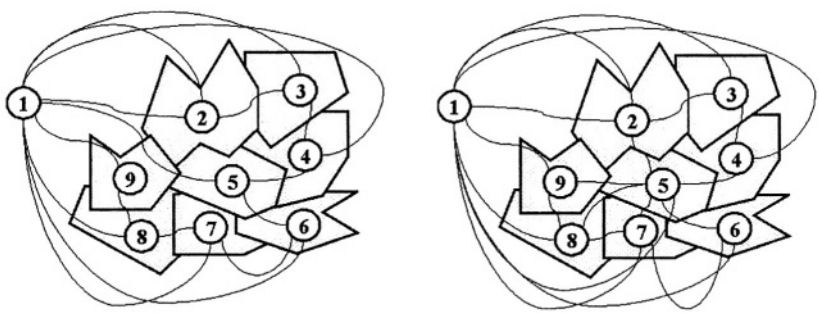
Fig. 5.6. Neighborhood graph of a homogeneous decomposition into tessellations.

A basic problem is linked to multiple neighborhoods. A solution to this problem of finding an adequate abstraction for a decomposition is to represent it by a *connection graph.*

Connection Graph

Connection graphs are planar graphs which encode topological neighborhood relations between the polygons of a tessellation. Therefore, pSRTs can be reduced to a set of connection graphs (representing neighborhood relations at different levels of granularity), which are interconnected by a decomposition tree (representing the hierarchical partonomy of reference units).

Definition 5.9
The connection graph of a homogeneous decomposition by tessellation with neighborhood graph $\mathcal{G}_{\mathcal{N}} = (V_{\mathcal{N}}, E_{\mathcal{N}})$ *is a graph* $\mathcal{G}_{\mathcal{C}} = (V_{\mathcal{C}}, E_{\mathcal{C}})$ *together with the combinatorial embedding of* $\mathcal{C}$ *in the plane.* $V_{\mathcal{C}} = V_{\mathcal{N}} \cup \{E\}$ *where* E *is the exterior, unbounded polygonal region.* $E_{\mathcal{C}}$ *contains an edge* (P_i, P_j) *for each connected sequence of polygon edges that* P_i *and* P_j *share. The combinatorial embedding of* $\mathcal{C}$ *consists in the circular ordering of the edges from* $E_{\mathcal{C}}$ *at each vertex from* $V_{\mathcal{C}}$.



(a) Connection graph with ordering information

(b) Circular sequence of neighbors of polygon 5

Fig. 5.7. Connection graph representation of a decomposition by tessellation.

Figure 5.7 shows the connection graph $\mathcal{C}$ of a homogeneous decomposition by tessellation D. Each polygon from D is represented by a vertex from $\mathcal{C}$. In addition, there is the node 1 representing the external polygonal region. The edges from $\mathcal{C}$ which are incident with a vertex are easily obtained together with their circular ordering by scanning the contour of the corresponding polygon (figure 5.7a. For polygon 5 the following circular sequence of neighbors is obtained (see additional edges in figure 5.7b): 1, 2, 4, 6, 7, 9. Note that polygon 2 which shares three edges with 5 appears only once because the three edges

are connected. The same holds for polygon 4, 6, and 9 with two edges. They have the same representation as polygon 7, which only has one edge connected to 5. As the example shows, the connection graph is a multi-graph in which several edges can join the same pair of vertices.

The connection graph representation supports a number of graph-theoretical operations which can be used to draw inferences about spatial relevance (spatial neighborhood). For example, *polygonal footprints spatially relevant to a given footprint* can be determined by breadth-first search in the connection graph. Another example is, to determine *polygonal footprints spatially relevant to a given set of footprints.* This can be done by the use of ordinal information given by the combinatorial embedding.

5.2.3 Place Name Structures

To overcome the mentioned limitations in reasoning capabilities we have developed a new representation scheme for place names in the form of place name structures (PNS). Place name structures provide representations of the regional extent of spatial objects in geographic space. They are formalized with the help of both topological and partonomic relations to reference units provided by the pSRT.

The polygonal standard reference tessellations can be seen as an analogon to the common vocabulary described in section 4. This way, we are able to integrate heterogenous place name structures. They provide means to extensionally define approximate locations within a reference tessellation.

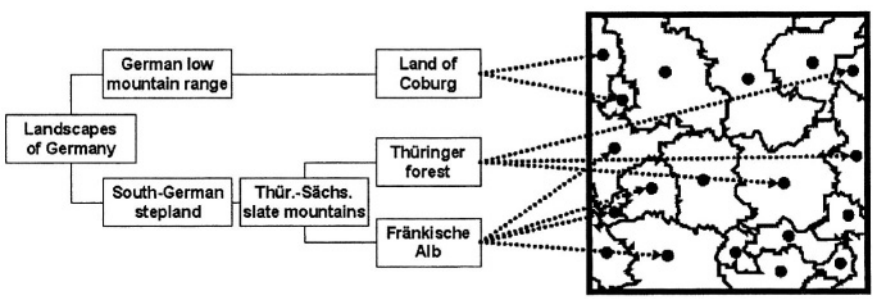


Fig. 5.8. Place name structure with binary projection on pSRT.

Figure 5.8 shows an example of such an extensional definition. The place names *Land of Coburg, Fränkische Alb,* and *Thüringer forest* are extensionalized onto a polygonal standard reference tessellation. Please note that *any* place name can be extensionalized. This means that also the *South-German stepland* can be mapped onto the tessellation. In addition, we can derive *intensional* information out of the PNS using the partonomic structure. This kind of spatial modeling allows us to define two relations: close-to and part-of.

One might think that PNS are some kind of geo-ontology. We can argue that PNS is a specific conceptualization of geographic space, however, these "concepts" are rather instances. Also, PNS should not be confused with "ontologies and geographic kinds" [104], where the authors argue that geographic objects are not merely located in space but are tied intrinsically to space. Furthermore, they state, that this means their spatial boundaries are in many cases the most salient features for categorization. Our place name structures are, as mentioned earlier, an extension of gazetteers and represent a specific conceptual view.

5.3 Spatial Relevance Reasoning

The requirements in section 5.1.4 show that a combination of neighborhood information and hierarchical information such as administrative units is useful. The definitions 5.8 and 5.9 allow a new type of reasoning: spatial relevance reasoning.

Spatial relevance reasoning is based on the assumption that the relative spatial relevance of two place names A and B is inversely proportional to both their (spatial) distance $dist_s(A, B)$ in the connection graph, and their (partonomic) distance $dist_p(A, B)$ in the decomposition tree. The easiest way to calculate a spatial relevance is to apply a linear function.

Definition 5.10
A spatial relevance $relev_s$ *is a cumulation of the distances* $dist_s$ *and* $dist_p$*. The weighting factor* α *allows to bias the spatial relevance:* $relev_s = \alpha\, dist_s(A, B) + (1 - \alpha)\, dist_p(A, B)$.

The weighting factor α can be used to bias the computation either towards the evaluation of true spatial distance (i.e., an qualitative approximation of Euclidean distance), or distance within the partonomy (i.e., within a context-dependent hierarchy).

If place name structures use the same reference tessellation or frame of reference, the integration of multiple place name structures can easily be achieved.

Definition 5.11
Let P_q *be a place name with* $P_q \in PNS_1$ *(q stands for query) with a spatial footprint* SFP_q *consisting of a set of reference units that belongs to a standard reference tessellation* $\{r_1, \ldots, r_n\} \in pSRT$*. Then a distance field* F_{dist} *can be computed in the connection graph* $\mathcal{C}_L$ *based on the pSRT at granularity level L.*

Based on F_{dist}, the spatial distance $dist_s(P_x, P_q)$ of a place name P_x that belongs to an arbitrary place name structure PNS_x can be computed, provided its spatial footprint SPF_{P_x} can be normalized to contain only reference units that are part of $\mathcal{C}_L$.

Definition 5.12
The partonomic distances $dist_{pi}(P_i, P_q)$ *for all place names in* PNS_x *are computed based on the partonomy encoded in* PNS_x*. Each node on the path to the top is assigned a partonomic distance of* $dist_p(P_0, P_n) = dist_p(P_0, P_{n-1}) + 1$*. For all nodes in the non-traversed sub-trees under P,* $dist_p$ *is set to* $dist_p(P_0, P_n)$*.*

Starting from the first common parent node P_0 of all place names $P_i \in PNS_x$ that share a minimal spatial distance to P_q, the hierarchical partonomy of PNS_x is recursively traversed to the top node. Using this metric, we can compute the spatial relevance of any place name P that belongs to an arbitrary place name structure PNS_x relative to a query location P_q. We will demonstrate the performance of spatial relevance reasoning in the following subsection.

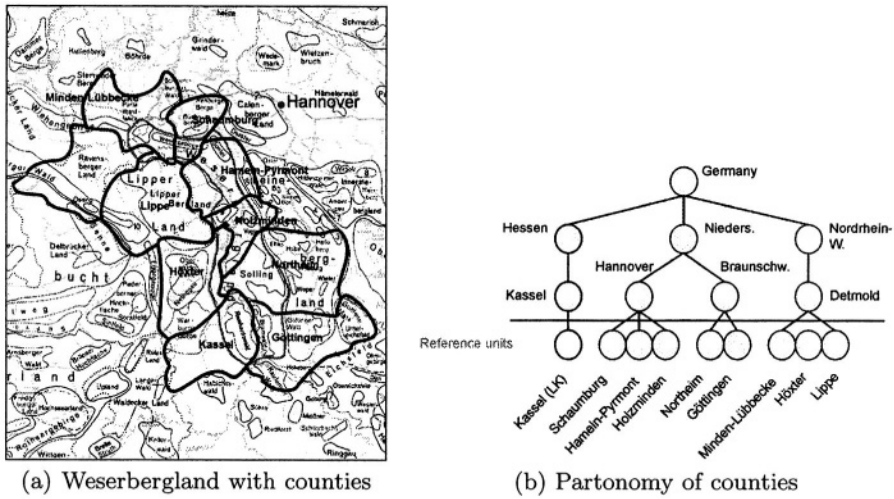
5.4 Example

We will describe three different scenarios. Firstly, we will show that spatial relevance reasoning on the polygonal standard reference tessellation pSRT is possible. This means that the pSRT does not only serve as a de facto "common vocabulary", Semantic Web users can also annotate their data with the help of these geographical terms. Secondly, we will show the reasoning capabilities with place name structures. Last, we demonstrate the integration capabilities between two or more place name structures.

Reasoning with Reference Units

Figure 5.9 shows an extraction of the map of landscapes of Germany. We see the fuzzy geographical area "Weserbergland" marked as light gray in figure 5.9a). The figure also shows those counties that are covered or partly covered by the Weserbergland area (thick black lines). Figure 5.9b shows the partonomy of theses counties accordingly.

The BUSTER prototype (see also 7 on p. 125) follows the concept of spatial relevance reasoning as described above. Therefore, the user is able to determine whether they want to put emphasis on neighborhood or hierarchical information, by means of the weighting factor α. When choosing 'neighborhood' only and looking for a hotel in the county of *Holzminden* for example, a user would only get information items if they are annotated with the spatial term Holzminden. If we choose a wider radius, which is an additional feature for querying, we would also get those information items annotated with the direct neighbors of the county Holzminden. The following holds: the wider the radius the bigger the chance and higher the number of hits for a query. However, the spatial relevance equation 5.10 also makes sure that the possible answers are ranked higher the closer the information item is (to the spatial query).



(a) Weserbergland with counties (b) Partonomy of counties

Fig. 5.9. Landscapes of Germany.

If 50% neighborhood and 50% partonomic importance have been set with a radius set to 1, we get information items that are located in 'Hameln-Pyrmont' and 'Schaumburg'. Figure 5.9b shows why: on the reference units level we can see that Holzminden, Hameln-Pyrmont, and Schaumburg are close together, being part-of Hannover. Since the radius is set to 1 both the neighborhood distance and the partonomic distance just cover those three counties. If we set the radius to 2, the next hierarchical level would be considered, which is the state of Niedersachsen in this case. This implies that the reasoner would also be able to find information items in 'Braunschweig', 'Northeim' and 'Göttingen' because they are part-of Niedersachsen.

If we would choose the hierarchical relevance only, the neighborhood information is not considered. Suppose we are looking for an information item in the county of Göttingen (this could be the closest school for example). With a small radius of one or two we would get information items located in 'Northeim', 'Holzminden', 'Schaumburg', 'Hameln-Pyrmont' on the lowest level and 'Braunschweig' and 'Niedersachsen' on the next higher levels (fig. 5.9b). Please note that the direct neighbor 'Kassel (Landkr.)' is not considered. Kassel (Landkr.) is part-of another state, namely the state of 'Hessen'. Therefore, the hierarchical distance is much higher than the horizontal distance. Thus, no answers to our query would be returned, which is correct as schools are tied to states.

Reasoning with Place Name Structures

For this application, we have chosen an area in the north-eastern part of Bavaria in Germany. In order to show the effect of our approach we first need

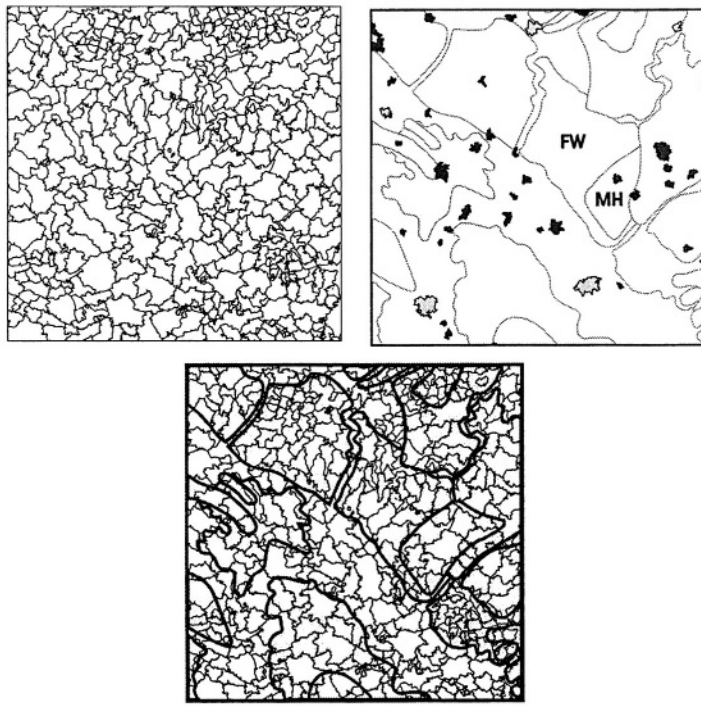


Fig. 5.10. Required for our example: (a) the spatial reference units as tessellation, (b) the place name regions, and (c) a combined schematic view of both.

a polygonal tessellation as spatial reference units. In our example, we have chosen the boundaries of the counties, however, this could also be any other polygonal tessellation (e.g., zip codes). As place name regions we have chosen a digital map of landscape areas. These areas are vague, i.e., even experts fight over the exact boundaries of these regions (e.g., *Frankenwald (FW)*). Figure 5.10 shows (a) the tessellation and (b) the polygons of the landscapes.

5.10(c) shows both layers combined in a schematic view for better understanding. If we look for an accommodation in a certain area, e.g., the 'Frankenwald' our approach would map this place name region into discrete space (the reference units). This is done by the determination of the upper and lower approximation as described in [120]. Let's say the approximation starts with an arbitrary polygon, e.g., the landscape polygon of the *Frankenwald (FW).* Figure 5.11 a and b show both, the upper and the lower approximation of this polygon. The reasoner would now be able to derive the possible answers and rank them according to the users specifications. At the moment, the reasoner considers the upper approximation only, however, a more precise mapping is currently under development. The reasoner also finds information items that are annotated with the county names directly. Hence, the user is able to

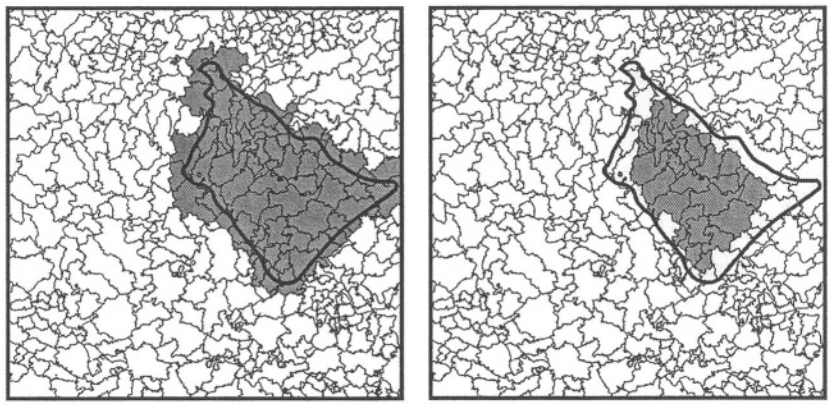


Fig. 5.11. (a) Upper and (b) lower approximation for the place name region *Frankenwald.*

type in more common names such as widely knows landscapes (Frankenwald) rather than specify non-intuitive reference units. This shows the flexibility of our approach: the user can type in place names but the knowledge engineer who annotates the information items for the Semantic Web is able to use both place names and county names.

Mapping Between Two Place Name Structures

Another important, if not the most important, feature of our approach is the ability to compute the spatial relevance of any place name P that belongs to an arbitrary place name structure PNS_x relative to a query location P_Q. Figure 5.12 schematically shows the mapping between two place name structures. The spatial relevance of a place name P in PNS_2 (modeling natural regions in Germany) with respect to a query location P_Q in PNS_1 (modeling the distribution of the regional offices of a firm) is computed as function of the geographic location of P and its position within the hierarchical structure of PNS_2.

This means that we would be able to find eligible information items annotated by different users using their own place name structure (regional offices of a company are a good example). However, this hold only if the Comprehensive Source Description (see section 3) points to the same reference tessellation. This is analogue to the common vocabulary described in section 4.

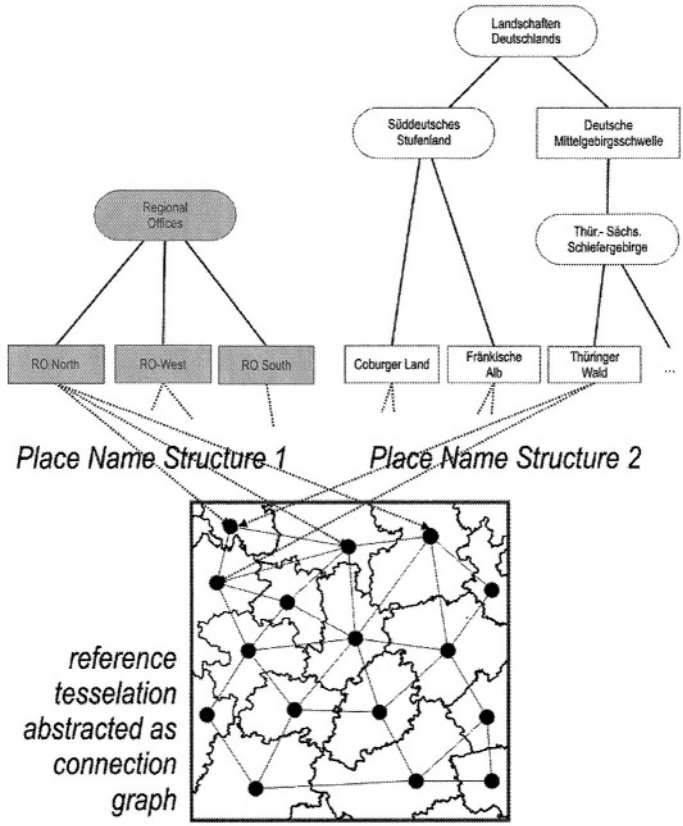


Fig. 5.12. Two place name structures and their mapping onto the standard reference tessellation.

This page intentionally left blank

6

Temporal Representation and Reasoning

This section describes the requirements which must to take into account with regard to the annotation and querying of temporal information sources. In the following, we discuss how our qualitative abstraction of time is represented. Temporal relevance is an important feature for the calculation of overlapping time periods with unknown boundaries. This is discussed in the following subsection. We will also describe the development and implementation of new reasoning components and demonstrate the performance of this approach with examples.

This chapter summarizes ideas that have been published elsewhere [125]. The representation and reasoning features described in this chapter are also based on the results of a masters thesis [52] that I initiated and supervised. This part of the BUSTER approach has been introduced lately [125].

6.1 Requirements

Annotation and retrieval of temporal information should be more flexible, comfortable, and improve situations in practise (e.g., with the help of colloquial terms such as Easter 2003). Both the knowledge engineer and the user should have several options to annotate or retrieve information for their purpose.

6.1.1 Intuitive Labeling

The most important requirement is the option to label time intervals with intuitive names. These names should be published and can therefore act as reference intervals for further internal or external use. However, typical country-dependent characters and unusual features have to be considered. We therefore restrict these names using existing standards such as UNICODE [115] for characters and the XML standard for names [135].

6.1.2 Time Interval Boundaries

Boundaries of time intervals should be flexible and have therefore various specifications. It is necessary that the boundaries on both sides of a time interval can differ. These different types are *exact, fuzzy, persistent,* and *unknown.* All possible combinations should be possible.

Exact Boundaries of Time Intervals

Exact boundaries represent a known, exact beginning and end. They are therefore the most simple case. An example for an exact boundary is the summer break in school: the vacation in the city of Bremen in 2002 started on the 20th of June and lasted until the 31st of July. The W3C offers a known encoding scheme [134], however, this scheme only considers time between the years 1 and 9999 of the Gregorian calendar. If we consider having information sources describing Julius Caesars moves in the years BC, we will have a problem. Therefore, the encoding scheme has to be extended.

Fuzzy Boundaries of Time Intervals

There are cases when a boundary is known but cannot be exactly determined. The beginning of an interval can then be described with the "earliest" and "latest" beginning. The same holds true for the end of an interval. This type of boundary can be chosen if more than one "official" opinion about a certain boundary, e.g., if recognized experts opinions differ. This can occur often when using common terms such as the "Middle Ages" and are therefore important. We usually have a good impression of time interval covering the Middle Ages but, we cannot exactly determine the beginning and the end.

Persistent Boundaries of Time Intervals

Persistent boundaries can appear if a given boundary is unrestricted, i.e., the interval still exists or the interval is already valid. This type of boundary is necessary for the end of an interval, when an end to the interval is not reached and cannot be determined or estimated. We see this phenomenon in scientific programs: a time interval with a defined beginning and an undefined end. Sending satellites or probes in the universe or carrying out a long-term observation is another typical example. When also note this for the beginning of an interval. We could have a time interval that begins before the *annotated* time period. Instead of using the minimal value for the lower boundary, we can use the persistent type.

Unknown Boundaries of Time Intervals

Unknown boundaries are necessary if no dates for the beginning or the end of a time interval are known. With this type of boundary it is also possible to define intervals where only one boundary (either the lower or upper boundary) is known. However, even if both sides are unknown, there is still the option to use this interval for statements about qualitative relations regarding other intervals. The delimitation to fuzzy or persistent boundaries is often not clear and is the discretion of the knowledge engineer. If we know the date of birth of a person but do not know the date of death, the use of an unknown boundary for the end of the time interval is obvious. If on the other hand existing documents (e.g., letters, official notifications) give proof at which time the person was alive and at which time that person died (also documents), we can use fuzzy boundaries. If that person is still alive, a persistent boundary could also be used. An interval with two unknown boundaries is a special case and states basically that there is a time interval only with a given name. If we use this interval with explicit relations (see below) we can make further statements.

6.1.3 Structures

An interval can be based on another interval, can be self-defined or imported. Exact and fuzzy boundaries for the beginning or the end of intervals for instance can be used to determine the exact end of an interval with the help of the beginning of another interval. Time points are used in order to carry out this operation. Therefore, functions are needed to extract these significant time points from the intervals. Examples for these functions are *beginning_of, end_of, earliest_beginning_of, latest_beginning_of, earliest_end_of, latest_end_of.*

An example for the different operations is the time interval “Middle Ages”, which historically cannot be exactly determined. However, there are existing events that can be used for the beginning or the end (see figure 6.1). Implicit qualitative relations exist through structures which are build upon each other (see relation *younger* that holds between “West-Roman Empire” and “Reign of Karl the Great”). These implicit relations are at the users disposal, together with the explicit relations, and contain the same expressive power (e.g., transitivity).

6.1.4 Explicit Qualitative Relations

Making statements about relations between intervals when using persistent or unknown boundaries should also be possible. This can be of value when we do not focus on exact or fuzzy boundaries but need to use the interval for qualitative relations. Consider the following example: firstly, we describe and order historic epoches. Secondly, having described the other intervals such as

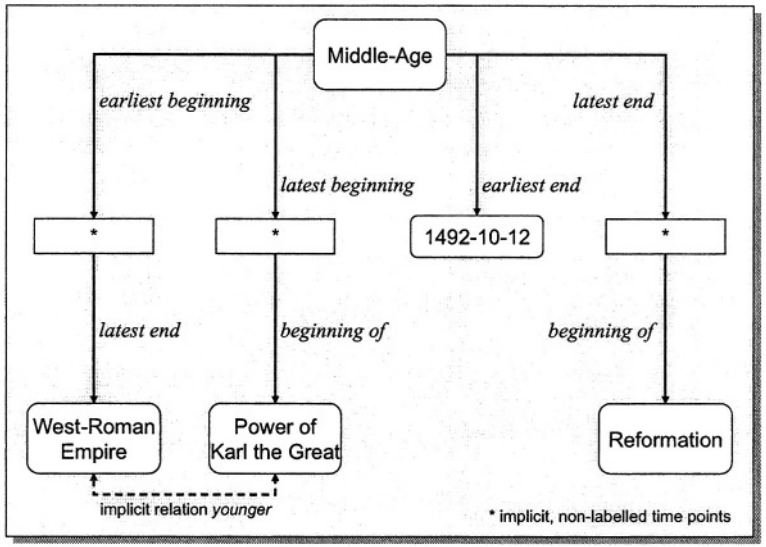


Fig. 6.1. Interval structure, after Pitz (2002) and Giesenberg (2002).

government times, CVs, travel times etc. using the epoches intervals, we are able to derive temporal relations between the other intervals.

As already mentioned, the Dublin Core Metadata Initiative has made a suggestion for temporal annotation (DCMI Period). The required features however, are only partly covered when using their coverage.Temporal format. Therefore, new concepts and methods must be developed. When comparing qualitative temporal approaches that are based on intervals such as Allen's relations (see section 2.3 on page 25) we see that they require exact boundaries. Intervals with fuzzy, persistent or unknown boundaries are not considered. Also, structures are far more complex with Allen's approach because they can only be implicit and are therefore computational expensive. Allen's time logic can therefore act as a fundamental theory, which partly covers the mentioned requirements.

6.2 Representation

6.2.1 Period Names

In the following, we present a new concept which we call *period names.* They allow the qualitative modeling of time and take the mentioned requirements for annotation and retrieval into account. Since we are dealing with annotation and retrieval for the Semantic Web, we use the XML notation to define the concepts and sub-concepts. XML as a description language offers the advantage to use its internal reference system, which is useful for both modeling and implementation. In particular, the construction of period name structures

is easier and more comfortable. XML notation is also the basis for the reasoning components, which are discussed in section 4.2.3. However, we could also use other notations to show the representation (e.g., graphs).

The use of XML is not mandatory, however, we concentrate on this language with regard to the Internet. Therefore, we restrict the language and use the XML standard for names [11] for our underlying model. This standard requires that XML names consists only of letters and numbers. Special characters such as %, $, & or spaces are not accepted. However, the dot (.), the dash (-) and the underscore (_) are exceptions.

Definition 6.1 (PeriodName)
*A period name consists of a header and a body. The header consists of the keyword **periodName** and an attribute id, which labels the name of the period. The body consist of the definition of boundaries and relations.*

Here are two examples for the description of a periodName in XML notation.

Example 6.1

```
a)  <periodName id="Label">
      <!-- Definition of boundaries -->
      ...
      <!-- Definition of relations -->
      ...
    </periodName>

b)  <periodName id="AntiqueTime"/>
```

6.2.3 Boundaries

The most important property of a period name is its expansion. The model contains only intervals, which are non-empty and consist of more then one time point. Therefore, the start point must lie before the end point.

The basis of boundaries are period structures, which are constructed intervals using point structures (as described above). These point structures are bounded and discrete. We can assume a continuous time stream with discrete, ordered values. The minimal time unit is exactly one millisecond and all time points can be ordered and compared because of the linearity.

Issues about the accuracy of time intervals, which occur due to the discrete model, must be considered. For instance, we could have information that belongs to a century or year in historic time. Also, information such as months, days or hours that belong to daily news have to be taken into account. Computer interactions require even more accuracy, usually up to seconds or milliseconds. Our model represents time with millisecond accuracy which is also supported by ISO 8601 and W3C-DTF. Even though this level of accuracy is not always necessary, it is not a disadvantage. Fuzzy boundaries for example, can be used to define boundaries where we do not need exact time points based on milliseconds.

Definition 6.2
The temporal range $R^t = [B, E]$ *consists of time points between the beginning B and the end E of the range. B is the time point 01.01.9999, 12:00am, 0 seconds and 0 milliseconds B.C. and in the following is denoted by -9999 and E is the time point 31.12.9999, 11:59pm, 59 seconds and 59 milliseconds and in the following is denoted by +9999. The year zero does not exist.*

For our definitions, two additional sets are necessary:

Definition 6.3
$\mathcal{P}$ *is a set of negative and positive persistent boundaries,* $\mathcal{P} = \{\mathcal{P}^-, \mathcal{P}^+\}$.

Definition 6.4
$\mathcal{U}$ *is a set of unknown boundaries.*

Exact Boundaries

Exact boundaries are used if a time interval has a known or exactly defined expansion. Starting points and ending points are defined by exactly one time point. The definition can be accomplished in four different ways:

Definition 6.5
Start and end points are defined explicitly by single time points $t_{begin} \in [B, E]$ *and* $t_{end} \in [B, E]$ *with* $t_{begin} < t_{end}$. *A time point is defined by a millisecond.* t_{begin} *describes the start of a period and* t_{end} *the end of that period.*

Lemma 6.1
Both time points are included, thus, the shortest time period is two milliseconds ($t_{begin} + 1 = t_{end}$).

The following example in XML notation describes a meeting on January 16, between 10 and 10.30am.

Example 6.2 (Meeting on January 16, between 10 and 10.30am)

```
<periodName id="Meeting">
  <begin>
    2003-01-16A10:00:00.000-00:00
  </begin>
  <end>
    2003-01-16A10:30:00.000-00:00
  </end>
</periodName>
```

Definition 6.6
Start and end points are defined by another existing time period. The start and end point can be single time points $t_{begin} \in [B, E]$ *and* $t_{end} \in [B, E]$ *or fuzzy boundaries. References and structures which are constructed from these, need the following keywords:* beginOf, endOf, beginfOf, endfOf.

This example denotes that the earliest begin of the Middle Ages (MA) is the end of the West-Roman empire (WRE).

Example 6.3 (Earliest begin of the MA is the end of the WRE)

```
<periodName id="middle-ages">
  <beginf>
    <endfOf ref="West-Roman_Empire"/>
  </beginf>
</periodName>
```

The actual time is important, especially when formulating a query. Examples are: “the last two weeks” or “the next 24 hours”.

Definition 6.7
The keyword ***now*** *is used for actual time points* $t \in [B, E]$. *‘Now’ is available with an accuracy of a millisecond and can be combined with the* begin/end-*attribute* offset *to define periods relative to the actual time.*

The following example shows the last minute from an actual time point.

Example 6.4 (Last minute from now on)

```
<periodName id="last_minute">
  <begin offset="-60000">
    <now/>
  </begin>
  <end>
    <now/>
  </end>
</periodName>
```

Relative periods from the actual time are important but are not sufficient enough to describe concepts such as “today” or “this year”. Also, periods that occur regularly such as “Easter” or “Christmas” need to be considered. Formulas can be defined to describe these situations.

Definition 6.8
dformula *denotes a formula that returns a certain time point* $t \in [B, E]$. *The return value can be used directly for* begin *or* end.

Definition 6.9
pformula *denotes a formula that returns a time period* $t_{begin} < t_{end}$ *with* t_{begin} *and* $t_{end} \in [B, E]$. *pformula can be used only after reference keywords as they represent anonymous periods, which can be referenced as labeled periods.*

The example shows a time period from the beginning of this year until midnight today:

Example 6.5 (Beginning of this year until midnight today)

```
<periodName id="since_beginning_of_year">
  <begin>
    <beginOf>
      <pformula name="thisyear"/>
    </beginOf>
  </begin>
  <end>
    <dformula name="midnight"/>
  </end>
</periodName>
```

Fuzzy Boundaries

It is useful not to use exact boundaries while modeling common or colloquial terms. Therefore, we introduce fuzzy boundaries as an extension of exact boundaries and are able to use the already established means for these boundaries: explicit dates, references, now, and formulas.

Definition 6.10
Let $t_{begin} \in [B, E]$ *and* $t_{end} \in [B, E]$ *be the start and end point.* Fuzzy boundaries *consist of two boundaries for both the start and end point.* $t_{beginf} \in [B, E]$ *is the earliest beginning and* t_{begin} *is the latest beginning for that time period. Accordingly,* t_{end} *denotes the earliest ending and* $t_{endf} \in [B, E]$ *the latest ending.*

Lemma 6.2
In addition, the following order holds: $t_{beginf} < t_{begin} < t_{end} < t_{endf}$.

Lemma 6.3
The time difference a *between* t_{beginf} *and* t_{begin} *therefore has the minimum of 1 millisecond. The maximum is arbitrary. The same holds true for the time difference* c *between* t_{end} *and* t_{endf}.

The following example shows the fuzzy boundary "begin of the Middle Ages":

Example 6.6 (Earliest and latest begin of the Middle Ages)

```
<periodName id="begin-middle-ages">
  <beginf>
    <endfOf ref="West-Roman_Empire"/>
  </beginf>
  <begin>
    <beginfOf ref="Reign_of_Karl_the_Great"/>
  </begin>
</periodName>
```

An extension for references is also needed: we recall the known constructs *beginOf* and *endOf*. They denote the "inner" boundaries (latest begin or earliest end) of a time period. The extension is needed for the "outer" boundaries *beginfOf* and *endfOf* (earliest begin and latest end) of a period. The difference between two time periods, which are defined by exact boundaries and fuzzy boundaries that have the same extent, is the calculation with regard to relevance (see section 6.3).

Figure 6.2 shows a graphical notation of fuzzy boundaries. Three time periods, each with two fuzzy boundaries show that the extent of "fuzziness" (the tolerance or width of the boundaries) can vary arbitrarily. Also, we can see that the outer boundaries of time period A meet B's and C's latest begin. These outer boundaries have referenced boundaries form B and C.

Fig. 6.2. Graphical notation of fuzzy boundaries: three time periods with fuzzy boundaries.

Persistent Boundaries

Persistent boundaries are necessary for two reasons: firstly, the start or endpoint of a time interval is before or after the range of the underlying model, i.e., before -9999 and after +9999. Secondly, a time interval could have a known exact or fuzzy beginning but an unknown end (or vice versa), e.g., the end of that time interval does have an open end in the future (long-term experiments). For both cases the keyword 'unlimited' is introduced.

Definition 6.11
P^- defines a boundary that is known or fuzzy, but before the beginning of the range, i.e., $P^- < B$. P^+ defines a boundary that is known or fuzzy but, after the end of the range, i.e., $P^+ > E$. The time point of a persistent boundary $P_t \in \{P^-, P^+\}$ consist of the keyword begin *or* end *followed by the keyword* unlimited *with the value* true *if the beginning or the end of the time interval is known but not in the valid range, i.e., $t_{begin} \notin [B, E]$.*

The following example shows an interval with two persistent boundaries:

Example 6.7 (A time interval with two persistent boundaries)

```
<periodName id="Label">
  <begin unlimited="true"/>
  <end   unlimited="true"/>
</periodName>
```

A time interval with two persistent boundaries cannot be distinguished from another time interval with two persistent boundaries. Therefore, only combinations with other intervals with other types of boundaries is reasonable.

Unknown Boundaries

If no information about a time interval is known or the time points are too vague, i.e., even fuzzy boundaries are not reasonable, another type of boundary is necessary: the unknown boundary. It can help for a qualitative modeling and reasoning with regard to other (known) time intervals.

Definition 6.12
An unknown boundary consist of the keyword begin *or* end *followed by the keyword* unknown *with the value* true. *The time point of an unknown boundary* $t \in U$ *is not known. An unknown boundary could be in the valid range* $t \in [B, E]$ *or is part of a persistent boundary* $t \in \{P^-, P^+\}$*, it is simply not known. By default, the boundary is set to unknown.*

The following example shows an interval with two unknown boundaries:

Example 6.8 (A time interval with unknown boundaries)

```
a)  <periodName id="Label">
      <begin unknown="true"/>
      <end   unknown="true"/>
    </periodName>

B)  <periodName id="Label"> </periodName>
```

Figure 6.3 shows the reason for the integration of unknown boundaries: the boundaries of the three time intervals are not known but we can see that qualitative propositions between these intervals do exist. They can therefore be of value for reasoning processes.

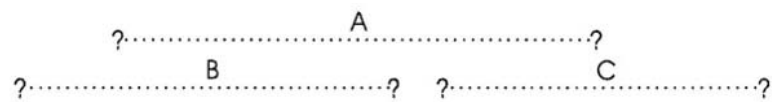


Fig. 6.3. Graphical notation of unknown boundaries: three time periods.

Combination of Boundaries

Using the same type of boundary for both start and end of a time interval is not useful. Time intervals with persistent boundaries especially develop their full potential in combination with time intervals having exact or fuzzy boundaries. Therefore, every possible combination of the described types of boundaries can be used while defining a period name. The user can also distinguish between subtypes of fuzzy boundaries such as explicit dates, references, “now”, and formulas to combine them with the other mentioned options.

6.2.4 Relations

If we use exact boundaries only, implicit relations between time intervals can be defined. A time interval could be completely covered by another time interval, overlap partly or one time interval could lay before the other. [2] identified 13 fundamental, distinguishable relations between time intervals. Freksa's critique that these are too exact and would imply too complicated models leads to the model of conceptual neighborhoods [35]. He introduced new concepts, which aggregate subsets of Allen's relations. These concepts are not as accurate, but they are easier to calculate with.

We can calculate relations from exact boundaries. We also can do this with fuzzy boundaries if we neglect the transition areas and only consider the outer time points. Therefore, the addition of new relations using these types of boundaries does not provide more information. Furthermore, it can only lead to redundancies or, even worse, to inconsistencies.

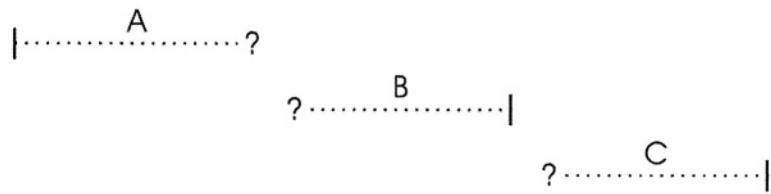


Fig. 6.4. Explicit relations.

However, new relations when dealing with single unknown boundaries or completely undetermined time intervals are important information sources. Consider the situation in figure 6.4: there are three time intervals, each with one known start or end time point. This leads to various sets of possible relations and we can assume that the relation between each pair is undetermined. Between A and B and A and C we can only eliminate $>$ (after) and *mi* (met-by) out of the 13 possible relations, the remaining 11 relations have to be considered:

$$A \ \{=, <, m, o, oi, d, di, s, si, f, fi\} \ B$$
$$A \ \{=, <, m, o, oi, d, di, s, si, f, fi\} \ C$$
$$B \ \{<, m, o, s, d\} \ C$$

If we would know that the end of time period A ends after the end of time period C (*A survives C, A sv C*) and add this piece of information to the system, the amount of possible relations could be reduced significantly:

$$A \ \{oi, di, si\} \ B$$
$$A \ \{sv\} \ C$$
$$B \ \{<, m, o, s, d\} \ C$$

Now, instead of 11 relations we only have three *oi, di, si* (overlapped-by, contains, started-by). According to Freksa, these remaining relations are also conceptual neighbors and can be aggregated into the concept "surviving contemporary of" (*sc*).

In order to specify a new explicit relation in a XML notation, the construct "relatedTo" is used. The attribute "ref" denotes another period name where the type of the relation is given by the attribute "type". Here is an example denoting the time period of the Middle Ages:

Example 6.9 (Middle Ages relations)

```
<periodName id="Middle-Ages">
  <!-- Definition of boundaries -->
    <begin unknown="true"/>
    <end   unknown="true"/>
  <!-- Definition of relations -->
    <relatedTo type="younger''     ref="Antiquity"/>
    <relatedTo type="survives"    ref="Antiquity"/>
    <relatedTo type="older"       ref="Modern_times"/>
    <relatedTo type="survivedBy"  ref="Modern_times"/>
</periodName>
```

In this example, both the starting and ending time points are defined as unknown. Then, we add the new relations (which are concerned with the relation of the outer time points) to those of other time intervals (e.g., younger as Antiquity).

6.3 Temporal Relevance

When using the temporal model to both annotate and retrieve information from the Web, the following question arises: how do we determine which data or information sources fit the query and to which degree? This can be summarized in the term of temporal relevance. Usually, the relevance is drawn on a scale between 0 and 1. The degree of relevance then mirrors the percentage of "fitness", i.e., 0 means that the found data do not fit the query at all, whereas, 1 means that the data fit the query with 100%.

After a thorough study of Allen's relations, we can group these into two main areas. One group consists of relations that consider disjunct time intervals only, i.e., *before* and *after*. The other group consists of relations that have an overlap of some kind (e.g., *during, contains*). However, there are two exceptions: *meets* and *met – by*. These can be seen as relations, which consider time intervals that are disjunct (by a millisecond) or overlapped (by one millisecond). For the following, we consider the latter and therefore group these two relations into the second area.

Furthermore, the temporal relevance can also be distinguished into two areas: (a) the distance and (b) the overlap of time periods. The latter can be refined to the consideration of distance between time points, namely the start and end time points of the considered time intervals.

6.3.1 Distance Between Time Intervals

The calculation between two time intervals *A* and *B* where the relation *A before B* holds true, is based on the distance between the end time point of *A* and the start time point of *B*. The length of the time interval is not relevant. Therefore, we can calculate the distance even if the other boundaries are unknown. Theoretically, 16 (4^2) combinations of two time intervals with different types of boundaries are possible. However, because we do not have to consider the types of boundaries that are at the start of *A* and the end of *B* we can reduce the number of combinations to ten (figure 6.5). The number of combinations from which we can draw conclusions is even lower if:

- one of the boundaries is unknown, we cannot make a comment about the relation and therefore we cannot calculate the distance. Four combinations out of the ten belong to this group (d,g,i, and j in figure 6.5);
- at least one of the boundaries is persistent, no distance can be calculated because one time interval is overlapping the other. Four combinations out of the ten belong to this group (one belongs to both groups: i,c,f, and h in figure 6.5).

Thus, three combinations where we have exact and fuzzy boundaries are left and have to be further considered. In the case of two exact boundaries, the calculation is simple because we can use subtraction. In the case of at least one fuzzy boundary, we simply calculate the mean average value of the tolerance area, i.e., the mean average value between the inner and outer boundary and then use subtraction for the overall distance. Once we have the distance, we can norm this value in order to get a value between 0 and 1.

6.3.2 Overlapping of Time Periods

The calculation of relevance between two overlapping time intervals causes a new consideration: it is important to know which time interval is the reference time interval and which time interval is the comparer. Figure 6.6 gives us some insight into this problem: we can see that *A* and *B* as well as *A* and *C* are overlapping. However, from the viewpoint of *B, A* is more important because *A* covers *B* completely. On the other hand *B* is not as important for *A* because the degree of overlapping of *B* is smaller than the one of *C*.

In contrast to the process of calculation with regard to the distance where we could calculate with the two opposite time points, we have to consider all four boundary time points of the two time intervals. Theoretically, we have

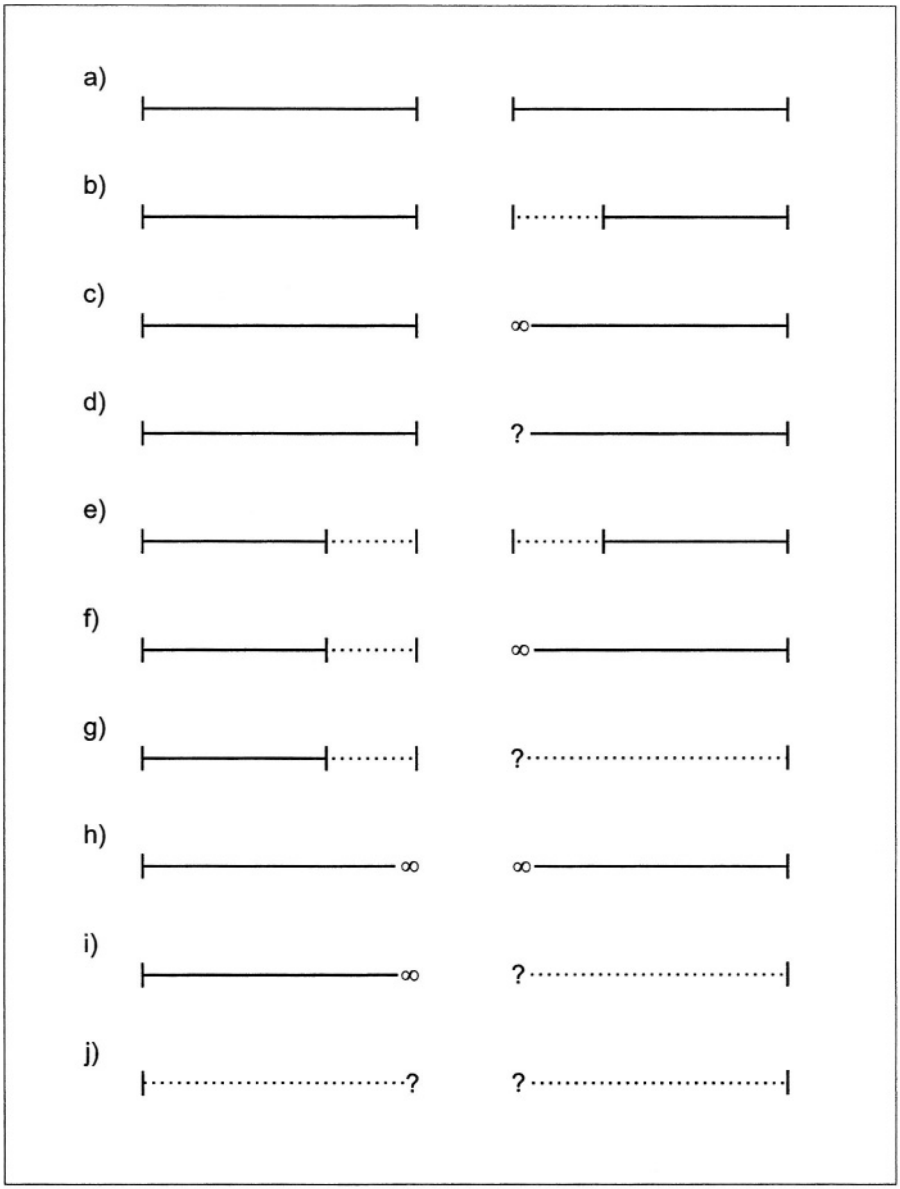


Fig. 6.5. Distance: the possible combinations of boundary types.

to consider 256 (4^4) combinations, which can be reduced due to symmetry drastically. None of the boundaries ought to be unknown since we cannot calculate any relevance. Also, persistent boundaries can be transformed into exact boundaries if the reference time interval has only exact or fuzzy boundaries.

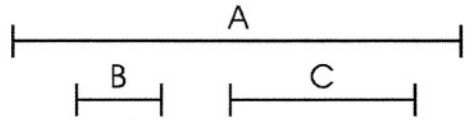


Fig. 6.6. Relevance as extent of overlapping.

The calculation of the relevance between two intervals with exact boundaries is straight forward: the length of the overlapping area can be related to the overall length of the reference time interval. If both intervals are identical, the relevance is 100%. The distance calculation with fuzzy boundaries must have a different result than the distance that would have been calculated using exact boundaries. Therefore, the width of the fuzzy area must have a significant influence on the result. Figure 6.7 shows the representation of fuzzy boundaries: the fuzzy area at the start of the time period (a) is the area between *beginf* and *begin*. The area between *begin* and *end* (b) is the area, which is certain, and the area between *end* and *endf* (c) is the fuzzy area at the end of the time period.

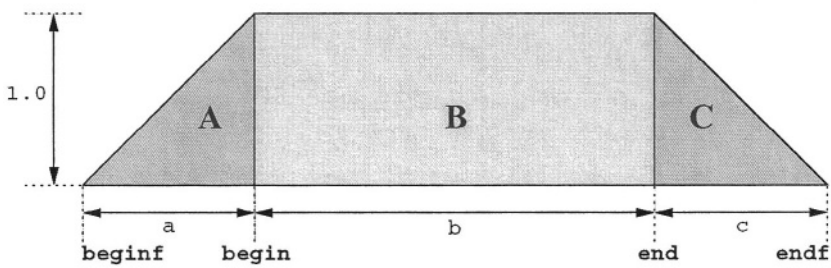


Fig. 6.7. Representation for fuzzy boundaries.

This representation follows the representation known as fuzzy set theory [146] where we have a moving transition of elements belonging to a set or not. The membership is described by a function that maps onto values between 0 and 1. Thus, we are able to represent common terms with fuzzy boundaries such as “warm” or “tall”. If we need to calculate the relevance of a time period (a possible answer to our query) with regard to a reference time period (our query), the overlapping area has to be determined. The overlapping area includes both fuzzy areas at the beginning and the end of the time periods and the “certain” area in the middle. After determining the overlapping area, we calculate the relation of the two time periods simply by dividing them: $\frac{A}{B}$ where A is the time period of the possible answer and B is the reference time period. The result is the temporal relevance for time period A with regard to time period B. Further details are described in [52].

6.4 Reasoning Components

We have introduced the concept "time period" for the abstract representation of time and time-based relations. Also, a simple algorithm for the calculation of temporal relevance has been described. Both representation and relevance are necessary to develop reasoning components that are described in the following. However, some assumptions and restrictions must first be made.

We have seen 30 relations between time intervals in total, 13 have been introduced by Allen, another 17 by Freksa (17 semi-interval comparisons of Allen's disjunctive sets). The more temporal cohesions a reasoner is able to process, the more powerful and efficient it is. The development of a temporal reasoner is currently undertaken (a first prototype has been finished) and we have started with the most important temporal relations with regard to the Semantic Web:

- older
- younger
- contemporary
- survives
- survived-by

The selection of these five relations is described in [52] in more detail.

In order to get conclusions based on the temporal model, new algorithms have to be developed. Allen used a constraint-based system to reduce the set of possible relations when adding new information. The system is also able to detect inconsistencies, however, the system is very limited. Therefore, we extend and modify Allen's approach in order to tackle the new temporal model (e.g., for fuzzy, persistent, and unknown boundaries, references). A particular feature is the co-existence of quantitative descriptions of periods and qualitative relations of such periods. In addition, with regard to the Semantic Web, it is imperative to detect inconsistencies.

6.4.1 Relations Between Boundaries

Considering the boundaries of time periods we can derive implicit relations. First, we have to compare the time points of those boundaries. If these time points are exact, we can order these and get three relations: (a) a time point does lay *before* another time point (<), (b) a time point does lay *after* another time point (>), and (c) the time points are the *same* (=).

Considering at least one fuzzy boundary is sufficient to compare the outer time points. This way, we take the maximum expansion of the time period into account and therefore simulate a time period with exact boundaries. This is possible due to the fact that the relations identified in the two groups of relevance (distance and overlapping) do not distinguish between exact and fuzzy boundaries. Thus, we also have the three relations $<, >, =$ for fuzzy boundaries.

Persistent boundaries cannot be mapped onto concrete time points due to the concept of the point structure (see 6.2). Therefore, numerous situations must be distinguished. First, persistent boundaries can appear in two ways: they are persistent with regard to the start of the time period (negative persistent) or persistent with regard to the end of the time period (positive persistent). For the comparison with exact time points, which are derived from exact boundaries or the transformation from fuzzy boundaries, and for the comparison with persistent boundaries the following theorems hold:

Theorem 6.1. *A negative persistent boundary truly lays a) before all exact time points* $t_n \in [B, E], n \in \mathbb{N}$ *of the time range and b) before all positive persistent boundaries.*

Theorem 6.2. *A positive persistent boundary truly lays a) behind all exact time points* $t_n \in [B, E], n \in \mathbb{N}$ *of the time range and b) behind all negative persistent boundaries.*

Theorem 6.3. *The relation between two positive or two negative persistent boundaries is undetermined.*

This corresponds to a intuitive notion of a time period, which is infinite far towards the past or the future. Therefore, we can determine three relations with regards to two persistent boundaries: $<$, $>$, and *unknown.* This is the basis for comparisons between time periods with regard to their position.

When considering time periods with unknown boundaries, only one relation with regard to another arbitrary boundary can be made: *unknown.*

Remark 6.1
No proposition can be made with regard to a position of an unknown, exact, fuzzy, or persistent boundary.

The proofs to the theorems 6.1-6.3 and the remark 6.1 can be done with the consideration of all possible cases. Figure 6.8 shows a time line with negative and positive persistent boundaries P^-, P^+, the temporal range denoted by $[B, E]$, and the unknown boundaries U.

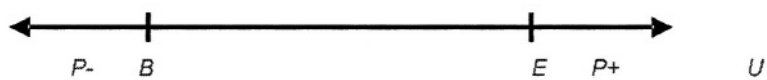


Fig. 6.8. Time line.

Proof 6.1 (-6.3)

$$\begin{aligned}
\forall s,t \in [B,E] \cup P^- \cup P^+ \cup U : s <^* t = \; & s < t \text{ if } \; s \in [B,E] \wedge t \in [B,E] \\
& \text{true if } \; s \in P^- \wedge t \in [B,E] \quad \text{(Theorem 6.1)} \\
& \text{true if } \; s \in P^- \wedge t \in P^+ \quad \text{(Theorem 6.1)} \\
& \text{true if } \; s \in [B,E] \wedge t \in P^+ \quad \text{(Theorem 6.2)}
\end{aligned}$$

$$\begin{aligned}
&\text{false if} \quad s \in [B, E] \wedge t \in P^- \\
&\text{false if} \quad s \in P^+ \wedge t \in P^- \\
&\text{false if} \quad s \in P^+ \wedge t \in [B, E] \\
&\text{false else} \quad \text{(Theorem 6.3, Remark 6.1)} \qquad \square
\end{aligned}$$

Thus, we are able to define four relations between time periods: $<$, $>$, $=$, and *unknown.* With this help, we can compare two time periods to derive their position relatively to each other.

6.4.2 Relations Between Two Time Periods

[35] introduced a method to determine the relations between two time periods by comparing abstract semi-intervals. These semi-intervals lay at the beginning and the end of the involved time periods, and are denoted as α and ω (or A and Ω).

This way, Allen's fundamental relations as well as Freksa's conceptual neighborhoods can be described easily. The procedure can be adapted for periods with exact boundaries by replacing the comparisons between semi-intervals with comparisons between margin points. The three relations used by Freksa ($<$, $>$, $=$) are also defined in this scenario.

Because we do not have to distinguish between exact and fuzzy boundaries, a transformation to an exactly defined time period with a maximal expansion can be made. Also, we are able to perform the same comparisons: it is not important whether or not the overlapping area belongs to a fuzzy area, what counts is the *existence* of a time period that is covered by the two time periods. Therefore, Freksa's definitions can be transformed onto exact or fuzzy boundaries.

If one of the time periods has at least one persistent boundary, two points have to be considered. First, the relation *equal* ($=$) is not defined. However, since none of the selected time period relations *(older, younger, contemporary, survives, survived-by)* is dependent on that particular relation the importance is marginal. Second, the position of two time periods can be *unknown.* In that case, no further propositions can be made.

Positive and Negative Sufficient Conditions

One of the benefits of our approach is that we can draw conclusions about relations of time periods, even if we have to deal with incomplete information. However, we need the necessary [52, page 41pp.] and sufficient conditions in order to draw conclusions. Table 6.1 gives an overview about the necessary and sufficient conditions for the selected relations. Suppose, we have two time periods (A and B) with only one exactly defined margin point per time period. Time period A has a defined end point and time period B has a defined

	necessary	sufficient positive	sufficient negative
older	$\alpha < A$	$\omega < A$	$\alpha \geq \Omega$
younger	$\alpha > A$	$\alpha > \Omega$	$\omega \leq A$
survives	$\omega > \Omega$	$\alpha > \Omega$	$\omega \leq A$
survived-by	$\omega < \Omega$	$\omega < A$	$\alpha \geq \Omega$
contemporary	$\alpha < \Omega \wedge \omega > A$	$\omega \leq \Omega \wedge \omega > A$	$\alpha \geq \Omega \vee \omega \leq A$
		$\vee\ \alpha < \Omega \wedge \alpha \geq A$	
		$\vee\ \alpha \leq A \wedge \omega > A$	
		$\vee\ \alpha < \Omega \wedge \omega \geq \Omega$	

Table 6.1. Necessary and sufficient conditions for the five selected relations.

beginning. The other margin points are unknown. In this situation, the check of the relation A *older* B would return nothing since the relation between the beginning of $A(\alpha)$ and the beginning of $B(\alpha)$ is not defined (see remark 6.1 above). On the other hand, we do know that the end of $A(\omega)$ is truly before B. Together with our fundamental demand that a beginning of an interval lays always before the end $(\alpha < \omega; A < \Omega)$ we can conclude that $\alpha < A$, i.e., the relation A *older* B is valid. The same holds true for B *younger* A.

This method checks a *positive sufficient* condition with $\omega < A$. However, there are also *negative sufficient* conditions. In the mentioned situation we can see that $\omega > \Omega$ does not hold: $A > \omega$ can be read from the position of the exact boundaries, $\Omega > A$ holds true by definition for all intervals. Also, due to the transitivity of $>$ the relation $\Omega > \omega$ holds true. Thus, the relation A *survives* B is therefore rejected by the negative sufficient condition $A > \omega$.

6.4.3 Relations Between More Than Two Time Periods

The comparisons between boundaries and two time periods enable us to make statements about cohesions between more than two time periods. Allen's composition table is a known approach for the concatenation of two relations. However, the restriction of the selected relations *older, younger, contemporary, survives* and *survived-by* make the construction for another composition table unnecessary.

Theorem 6.4. *The inverse relations rule themselves out* (older and younger; survives and survived-by), *all other combinations are possible, e.g.,* A ol B; A ct B; A sv B. *Further, we can aggregate the relations into two groups: (a) reflexive (6.1) and symmetric (6.2)* (contemporary) *and (b) non-reflexive (6.3), anti-symmetric (6.4), and transitive (6.5)* (older, younger, survives, survived-by).

$$\forall p \in P : (p\ ct\ p) \tag{6.1}$$

$$\forall p_1, p_2 \in P : (p_1\ ct\ p_2) \longrightarrow (p_2\ ct\ p_1) \tag{6.2}$$

$$\forall p \in P : \neg(p\ ol\ p) \tag{6.3}$$

$$\forall p_1, p_2 \in P : \neg(p_1 \; ol \; p_2 \wedge p_2 \; ol \; p_1) \tag{6.4}$$

$$\forall p_1, p_2, p_3 \in P : (p_1 \; ol \; p_2 \wedge p_2 \; ol \; p_3) \longrightarrow (p_1 \; ol \; p_3) \tag{6.5}$$

where P is the set of all time periods.

Also, we can show that a time period p_1 overlaps another time period p_2 if p_1 starts earlier and ends later (6.6); the invers relations hold correspondingly (6.7).

$$\forall p_1, p_2 \in P : (p_1 \; ol \; p_2 \wedge p_1 \; sv \; p_2) \longrightarrow (p_1 \; ct \; p_2) \tag{6.6}$$

$$\forall p_1, p_2 \in P : (p_1 \; yo \; p_2 \wedge p_1 \; sb \; p_2) \longrightarrow (p_1 \; ct \; p_2) \tag{6.7}$$

Proof 6.4

The statements 6.1-6.7 can be derived from the definitions of the relations about semi-interval comparisons and the implicit relations $\alpha < \omega$ (and $A < \Omega$) for each time period. This can be shown first for the relation contemporary:

$$\forall p \in P : (\alpha < \omega \wedge \omega > \alpha) \longrightarrow \forall p \in P : (p \; ct \; p) \quad \square \tag{6.8}$$

$$\begin{aligned} \forall p_1, p_2 \in P : \; & (p_1 \; ct \; p_2) \\ & \longrightarrow (\alpha < \Omega \; \wedge \; \omega > A) \\ & \longrightarrow (A < \omega \; \wedge \; \Omega > \alpha) \\ & \longrightarrow (p_2 \; ct \; p_1) \quad \square \end{aligned} \tag{6.9}$$

The relations older, younger, survives *and* survived-by *use semi-interval comparisons $<$ and $>$ exclusively, including their non-reflexivity, anti-symmetry, and transitivity:*

$$\forall p \in P : \neg(\alpha < \alpha) \longrightarrow \forall p \in P : \neg(p \; ol \; p) \quad \square \tag{6.10}$$

$$\begin{aligned} \forall p_1, p_2 \in P : \; & \neg(\alpha < A \; \wedge \; A < \alpha) \\ & \longrightarrow \forall p_1, p_2 \in P : \neg(p_1 \; ol \; p_2 \; \wedge \; p_2 \; ol \; p_1) \quad \square \end{aligned} \tag{6.11}$$

$$\begin{aligned} \forall p_1, p_2, p_3 \in P : \; & (p_1 \; ol \; p_2 \; \wedge \; p_2 \; ol \; p_3) \\ & \longrightarrow (\alpha < \alpha'; \wedge \alpha' < A) \\ & \longrightarrow (\alpha < A) \\ & \longrightarrow (p_1 \; ol \; p_3) \quad \square \end{aligned} \tag{6.12}$$

6.6 and 6.7 can be shown accordingly:

$$\begin{aligned} \forall p_1, p_2 \in P : \; & (p_1 \; ol \; p_2 \; \wedge \; p1 \; sv \; p_2) \\ & \longrightarrow (\alpha < A \; \wedge \; \omega > \Omega) \\ & \longrightarrow (\alpha < \Omega \; \wedge \; \omega > A) \\ & \longrightarrow (p_1 \; ct \; p_3) \quad \square \end{aligned} \tag{6.13}$$

$$\begin{aligned}\forall p_1, p_2 \in P : \; & (p_1 \; yo \; p_2 \; \wedge \; p1 \; sb \; p_2) \\ & \longrightarrow (\alpha > A \; \wedge \; \omega < \Omega) \\ & \longrightarrow (\omega > A \; \wedge \; \alpha < \Omega) \\ & \longrightarrow (\alpha < \Omega \; \wedge \; \omega > A) \\ & \longrightarrow (p_1 \; ct \; p_3) \quad \square \end{aligned} \tag{6.14}$$

We can see that we can derive the cohesions between multiple time periods without a complex composition table. The most important means are symmetry of the *contemporary* relation and transitivity of the *older, younger, survives,* and *survived-by* relation.

6.5 Example

The following example gives us an impression of the reasoning performance of the mentioned temporal approach. As a basis we choose a structure with the period names “antiquity”, “Middle Ages”, and “modern times”. We vary their boundaries in order to demonstrate the reaction of the underlying engine.

```
<?xml version="1.0" encoding="ISO-8859-1"?> <periodNames
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="periodNames.xsd">

  <periodName id="antiquity">
    <relatedTo type="older" ref="middle-ages"/>
  </periodName>

  <periodName id="middle-ages">
    <relatedTo type="older"       ref="modern-times"/>
    <relatedTo type="survives"    ref="antiquity"/>
    <relatedTo type="survivedBy"  ref="modern-times"/>
  </periodName>

  <periodName id="modern-times"/>
</periodNames>
```

Fig. 6.9. Example “antiquity, Middle Ages, and modern times”.

6.5.1 Qualitative Statements

Suppose the boundaries of the three period names are completely undetermined and only a few qualitative statements with regard to the relations between them are known. Figure 6.9 shows this situation in a XML notation. The reasoner transforms the situation in an internal graphical structure and derives eight relations besides the known period names. Four of them are already given by the user (USER), another four can be derived by symmetry

from *older* and *younger* as well as *survived* and *survived-by* (IMPLICIT). The reasoner shows the following output:

```
Parsing: ok.
-----
Transformation DOM->Internal Representation: ok.
-----
# of Periods found: 3
"antiquity"      [UNKNOWN,UNKNOWN]
"middle-ages"    [UNKNOWN,UNKNOWN]
"modern-times"   [UNKNOWN,UNKNOWN]
-----
OLDER={
  "antiquity"    "middle-ages"  (? ,USER ,UNKNOWN)
  "middle-ages"  "modern-times" (? ,USER ,UNKNOWN)}

YOUNGER={
  "middle-ages"  "antiquity"    (? ,IMPLICIT,UNKNOWN)
  "modern-times" "middle-ages"  (? ,IMPLICIT,UNKNOWN)}

CONTEMPORARY={}

SURVIVES={
  "middle-ages"  "antiquity"    (? ,USER ,UNKNOWN)
  "modern-times" "middle-ages"  (? ,IMPLICIT,UNKNOWN)}

SURVIVEDBY={
  "antiquity"    "middle-ages"  (? ,IMPLICIT,UNKNOWN)
  "middle-ages"  "modern-times" (? ,USER ,UNKNOWN)}
```

Each relation consists of origin (USER/IMPLICIT), the temporal relevance, and a validity status. The two latter are unknown at present and therefore, the '?' and the term UNKNOWN is given.

The next step is the expansion and verification of the internal model. The already known relations will be given again, however, if the verification process can verify the qualitative relations with the help of quantitative comparisons, these will be shown. In this case, the validity status is the same than the above, i.e., no quantitative comparisons can be made. Here is an extract of the output (note the expansion by the reasoner, e.g., the "antiquity/modern-times"-relation in OLDER):

```
Expand and Verify: ok.
-----
OLDER={
  "antiquity"    "middle-ages"   (? ,USER ,UNKNOWN)
  "middle-ages"  "modern-times"  (? ,USER ,UNKNOWN)
  "antiquity"    "modern-times"  (? ,REASONER,UNKNOWN)}
```

```
YOUNGER={
  "middle-ages"  "antiquity"     (? ,IMPLICIT,UNKNOWN)
  "modern-times" "middle-ages"   (? ,IMPLICIT,UNKNOWN)
  "modern-times" "antiquity"     (? ,REASONER,UNKNOWN)}

CONTEMPORARY={}

SURVIVES={
  "middle-ages"  "antiquity"     (? ,USER ,UNKNOWN)
  "modern-times" "middle-ages"   (? ,IMPLICIT,UNKNOWN)
  "modern-times" "antiquity"     (? ,REASONER,UNKNOWN)}

SURVIVEDBY={
  "antiquity"    "middle-ages"   (? ,IMPLICIT,UNKNOWN)
  "middle-ages"  "modern-times"  (? ,USER ,UNKNOWN)
  "antiquity"    "modern-times"  (? ,REASONER,UNKNOWN)}
```

At this point, the temporal model is expanded to its maximal extent (12 relations). So far, no inconsistencies have been found between the qualitative relations and quantitative boundaries. In addition, no inconsistencies have been detected between two or more qualitative statements. Thus, after verifying the consistency, queries can be formulated.

One example is the following: "Which period names do have a known relation with *Middle-Ages,* what kind of relations are these, and which temporal relevance do they have?" Here is the outcome:

```
relatedTo middle-ages:
  older:        [antiquity(?)]
  younger:      [modern-times(?)]
  contemporary: []
  survives:     [modern-times(?)]
  survivedBy:   [antiquity(?)]
```

6.5.2 Quantitative Statements

The second example consists of the same structure and periods but with determined boundaries at the beginning and the end. Figure 6.10 shows the details, please note that some of these boundaries reference already defined boundaries (e.g., endOfref="antiquity").

After parsing and transforming the input, the following list of period names including their explicit relations is found:

```
# of Periods found: 3
"antiquity"     [-UNLIMITED,-46388592000000]
"modern-times"  [-14830992000000,+UNLIMITED]
"middle-ages"   [-46388592000000,-14830992000000]
```

```
<periodName id="antiquity">
  <begin unlimited="true"/>
  <end>
    0500-01-01T00:00:00.000+00:00
  </end>
</periodName>

<periodName id="modern-times">
  <begin>
    1500-01-01T00:00:00.000+00:00
  </begin>
  <end unlimited="true"/>
</periodName>

<periodName id="middle-ages">
  <begin>
    <endOf ref="antiquity"/>
  </begin>
  <end>
    <beginOf ref="modern-times"/>
  </end>
</periodName>
```

Fig. 6.10. Example "antiquity, middle ages, and modern times with determined boundaries".

```
-----
OLDER={}
YOUNGER={}
CONTEMPORARY={}
SURVIVES={}
SURVIVEDBY={}
```

Since the internal format of date consist of the number of milliseconds to or from the beginning of the "JAVA-epoch" (January 1st, 1970, 12.00am), the exact boundaries are shown as big negative numbers. The persistent boundaries differ in the sign according to the direction of "leaving" the range: negative sign for the past and positive sign for the future. The list of explicit relations is empty because there are no explicit relations given. After expanding and verifying the model, the output is the following:

```
OLDER={
  "antiquity"     "modern-times"  (? ,REASONER,VALID)
  "antiquity"     "middle-ages"   (? ,REASONER,VALID)
  "modern-times"  "antiquity"     (? ,REASONER,INVALID)
  "modern-times"  "middle-ages"   (? ,REASONER,INVALID)
  "middle-ages"   "antiquity"     (? ,REASONER,INVALID)
  "middle-ages"   "modern-times"  (1.0 ,REASONER,VALID)}
```

```
YOUNGER={
  "modern-times"  "antiquity"     (? ,REASONER,VALID)
  "middle-ages"   "antiquity"     (? ,REASONER,VALID)
  "antiquity"     "modern-times"  (? ,REASONER,INVALID)
  "middle-ages"   "modern-times"  (? ,REASONER,INVALID)
  "antiquity"     "middle-ages"   (? ,REASONER,INVALID)
  "modern-times"  "middle-ages"   (? ,REASONER,VALID)}

CONTEMPORARY={
  "antiquity"     "modern-times"  (? ,REASONER,INVALID)
  "modern-times"  "antiquity"     (? ,REASONER,INVALID)
  "antiquity"     "middle-ages"   (? ,REASONER,INVALID)
  "middle-ages"   "antiquity"     (? ,REASONER,INVALID)
  "modern-times"  "middle-ages"   (? ,REASONER,INVALID)
  "middle-ages"   "modern-times"  (? ,REASONER,INVALID)}

SURVIVES={
  "antiquity"     "modern-times"  (? ,REASONER,INVALID)
  "antiquity"     "middle-ages"   (? ,REASONER,INVALID)
  "modern-times"  "antiquity"     (? ,REASONER,VALID)
  "modern-times"  "middle-ages"   (? ,REASONER,VALID)
  "middle-ages"   "antiquity"     (1.0 ,REASONER,VALID)
  "middle-ages"   "modern-times"  (? ,REASONER,INVALID)}

SURVIVEDBY={
  "modern-times"  "antiquity"     (? ,REASONER,INVALID)
  "middle-ages"   "antiquity"     (? ,REASONER,INVALID)
  "antiquity"     "modern-times"  (? ,REASONER,VALID)
  "middle-ages"   "modern-times"  (? ,REASONER,VALID)
  "antiquity"     "middle-ages"   (? ,REASONER,VALID)
  "modern-times"  "middle-ages"   (? ,REASONER,INVALID)}
```

All these relations are found by the reasoner. Those relations that could be proven within the process of expanding are marked as "VALID". On the other hand, those that could be proven "INVALID" are marked as such. Please note that the invalid relations do not imply any inconsistencies. These are *implicit* relations and are therefore not inconsistent for the internal representation. The implicit relations are determined with the help of the theorems give in section 6.4. Theorem 1 for instance can be used to derive "antiquity *older* middle-ages".

The reasoner also found two relations where the temporal relevance could be determined (*middle-ages survives antiquity* and *middle-ages older modern-times*). In both cases, we compare the overlapping time interval of the actual time period with the time interval that is given by the significant points for the actual relation. *Older* uses the start points and *survives* uses the end

points of the periods to compare. These time intervals are identical because the periods are standing in relation to *meets* or *met-by.* Therefore, a temporal relevance of 1.0 is calculated. The temporal relevance cannot be calculated if the time points are persistent or unknown.

The following examples are shorter and only the significant outcomes are shown.

6.5.3 Inconsistencies (Quantitative/Qualitative)

In order to demonstrate the behavior of the reasoner with regard to inconsistencies our former example will be extended by an explicit relation, which is in direct contradiction to the modeled boundaries: *middle-ages older antiquity.* The following demonstrates the output after parsing and transforming the given model:

```
# of Periods found: 3
"antiquity"     [-UNLIMITED,-46388592000000]
"modern-times"  [-14830992000000,+UNLIMITED]
"middle-ages"   [-46388592000000,-14830992000000]
-----
OLDER={
  "middle-ages"  "antiquity"    (? ,USER ,UNKNOWN)}

YOUNGER={
  "antiquity"    "middle-ages"  (? ,IMPLICIT,UNKNOWN)}

CONTEMPORARY={}

SURVIVES={}

SURVIVEDBY={}
```

The validity value is unknown at this point. After expansion and verification inconsistencies are determined. Theorem 1 ,e.g., proves *antiquity older middle-ages* and therefore contradicts *antiquity younger middle-ages,* which implicitly can be derived with the help of the temporal model *middle-ages older antiquity.* The following outcome shows the inconsistencies, which make the overall model invalid (the invalid inverse relations are not shown for better understanding):

```
["Middle-Ages"---OLDER-->"Antiquity"]
KnowledgeBase contains 1 invalid and 0 contradictory relations!
Shutting down...
```

Once it is known that the temporal model is not consistent, queries cannot be made because the correctness of the results cannot be guaranteed.

6.5.4 Inconsistencies (Reasoner Implicit/Qualitative)

Another example for inconsistencies is the contradiction between explicit qualitative relations and relations that are derived by the reasoner using quantitative knowledge. In order to demonstrate this, we modify our example slightly as shown in figure 6.11. The internal representation does not contain contra-

```
...
<periodName id="antiquity">
  <begin unlimited="true"/>
  <end unknown="true"/>
</periodName>

<periodName id="middle-ages">
  <begin>
    0500-01-01T00:00:00.000+00:00
  </begin>
  <end unknown="true"/>
</periodName>

<periodName id="modern-times">
  <begin unknown="true"/>
  <end unknown="true"/>
  <relatedTo type="younger" ref="middle-ages"/>
  <relatedTo type="older" ref="antiquity"/>
</periodName>
```

Fig. 6.11. Example "antiquity, middle ages, and modern times creating an inconsistency".

dictions in the beginning between the boundaries and the modeled relations because they relate to an undetermined period:

```
# of Periods found: 3
"antiquity"       [-UNLIMITED,UNKNOWN]
"middle-ages"     [-46388592000000,UNKNOWN]
"modern-times"    [UNKNOWN,UNKNOWN]
-----
OLDER={
  "middle-ages"   "modern-times"  (? ,IMPLICIT,UNKNOWN)
  "modern-times"  "antiquity"     (? ,USER ,UNKNOWN)}

YOUNGER={
  "modern-times"  "middle-ages"   (? ,USER ,UNKNOWN)
  "antiquity"     "modern-times"  (? ,IMPLICIT,UNKNOWN)}

CONTEMPORARY={}
```

```
SURVIVES={}

SURVIVEDBY={}
```

We do not know the beginning or the end of "modern-times". Therefore, we can neither prove nor disprove *modern-times older antiquity* or *modern-times younger middle-ages* and the resulting inverse relations. Thus, the validity value stays unknown. During the expansion using the marginal points we can derive implicit relations such as *antiquity older middle-ages* using theorem 1 (because of the transitivity of the *older*-relation knowing *modern-times older antiquity*). Accordingly, we can prove the inconsistency *modern-times younger middle-ages.* Here is the outcome of the reasoning process:

```
["antiquity"---OLDER-->"modern-times",
 "middle-ages"---YOUNGER-->"modern-times"]

KnowledgeBase contains 0 invalid and 2 contradictory relations!
Shutting down...
```

The additional given relations are consistent in this case, however, combining those with quantitative statements can prove the contradictions.

6.5.5 Inconsistencies (Qualitative/Quantitative)

In our last example we demonstrate the appearance of contradictions having qualitative models only. We modify the above mentioned example accordingly showing cycles (see figure 6.12). While constructing the internal representation no inconsistencies between relations and boundaries were found because the latter are not defined. The expansion and verification process, however, finds contradictions within all three relations due to the asymmetry of *older.*

```
...
<periodName id="antiquity">
  <relatedTo type="older" ref="middle-ages"/>
</periodName>

<periodName id="middle-ages">
  <relatedTo type="older" ref="modern-times"/>
</periodName>

<periodName id="modern-times">
  <relatedTo type="older" ref="antiquity"/>
</periodName>
...
```

Fig. 6.12. Example "antiquity, middle ages, and modern times with qualitative relations only".

```
["antiquity"---OLDER-->"modern-times",
 "middle-ages"---OLDER-->"antiquity",
 "modern-times"---OLDER-->"middle-ages"]

KnowledgeBase contains 0 invalid and 3 contradictory relations!
Shutting down...
```

The reasoner identifies all inconsistencies, which can help to evaluate and modify the temporal model in order to eliminate the contradictions. In our case, the relation *modern-times older antiquity* could be eliminated or changed to *modern-times younger antiquity.*

We have shown that the reasoning process is able to detect all possible inconsistencies of a temporal model, which is based on a period names structure. Inconsistencies could appear (a) between qualitative statements and defined boundaries, (b) between qualitative statements and derived implicit relations, and (c) between qualitative statements containing cycles. In addition, inconsistencies are labeled to simplify the correction of the model.

This page intentionally left blank

Part III

Implementation, Conclusion, and Future Work

This page intentionally left blank

7

Implementation Issues and System Demonstration

This section describes *some* of the issues that have been discussed and implemented with regard to the prototypical BUSTER system. The main reason for this is that the general functionality and applicability of our approach play a more important role than how the system is being implemented. However, this does not mean that the prototype is outdated with regard to the implementation issues. The prototype is based on an open client/server architecture (cf. [127]) and can be divided into two main parts: the so-called BUSTER-cluster on the server side and a BUSTER client.

The cluster part contains all the relevant modules necessary to guarantee the functionalities described in the sections before. Since we already have given details about the functionalities and also have given examples of how the system operates in practise, we give a short overview about the architecture of our implementation in the first subsection.

The remaining part of this section deals with the BUSTER client. We demonstrate the performance and also the look-and-feel of the prototype with some real-world examples. Some of them have been mentioned earlier, however, we mainly discussed what takes place on the server side.

7.1 Architecture

A BUSTER client can be started as a local application or as a java applet in a standard browser supporting Java Swing. The BUSTER client provides an ontology-driven user interface to specify queries and to present the results of the retrieval. Additional services such as automatic translation processes, if applicable, will be made available dependent on the result. The communication between the clients and the cluster is implemented via Remote Method Invocation (RMI).

The BUSTER cluster comprises several modules relevant for intelligent querying and semantic translation purposes: a BUSTER server, a database for CSDs and available domains, a web server, and terminological, spatial, and

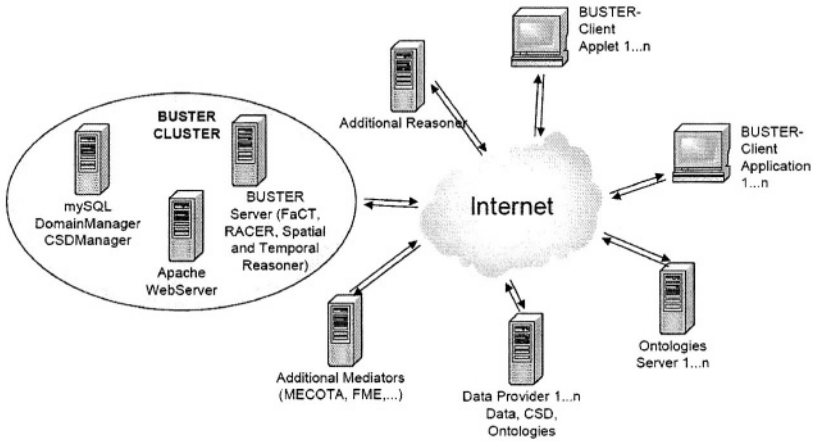


Fig. 7.1. BUSTER: system architecture.

temporal reasoning modules (see figure 7.1). Examples for the latter available on the Web are the FaCT system provided by the University of Manchester [59] and the RACER system provided by the University of Hamburg [47]. Both the spatial and the temporal reasoner are modules that have been developed and implemented within the BUSTER group.

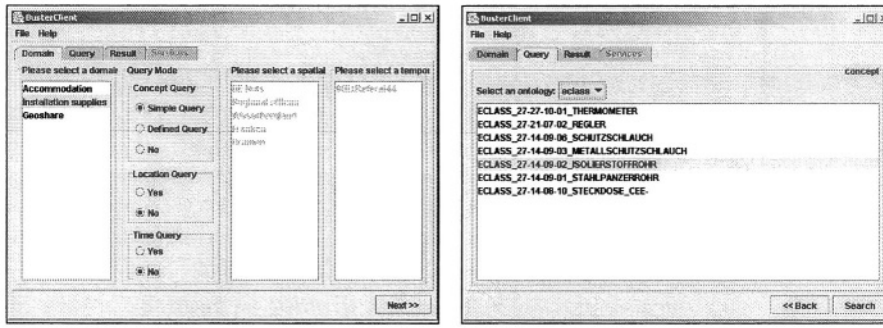
These modules within the BUSTER cluster fit the minimum requirements for terminology, spatial, and temporal queries, but the open architecture allows the use of arbitrary services for reasoning, translation or other tasks if needed.

An Apache Web server provides the platform for the applets and the server handles client queries depending on the users selection. It also controls the process of the query *(concept@location in time)* by retrieving domain specific information from a SQL-database via JDBC interface, downloading distributed CSDs and knowledge bases, and triggering reasoning services within or outside the BUSTER cluster.

7.2 Single Queries

Once an information source has been annotated with all the information needed, complex queries can be directed to the BUSTER system. As described before, BUSTER is based on terminological ontologies that have been modeled in advance. The system allows different types of queries:

- terminological queries,
- spatial queries, and
- temporal queries.



(a) BUSTER-Client start page: choosing the domain (or application area)

(b) Example for a simple concept query

Fig. 7.2. Start page and example for a simple concept query.

In addition, the possible combinations of these queries can be selected. Thus, it is possible to query the terminological part of the system without taking the spatial and temporal part into account. On the other hand, the query type *concept@location in time* is also possible, e.g., when looking for a hotel in a certain area and a certain time. The following subsections describe the most important query types.

7.2.1 Terminological Queries

The terminological query can be divided into two parts, namely a *simple concept query* and a *defined concept query*. In case of a simple concept query, the user has to choose a specific ontology. This makes the query simpler for a user to understand, but assumes that the user knows at least one concept from the ontology. Simple concept queries are fast, but not always expressive enough.

To overcome these problems we can use the defined concept query. On the base of the given common vocabulary, the user is able to define a concept that fits his vision of a concrete concept. A defined query is more complex to build, but it is much more unrestricted.

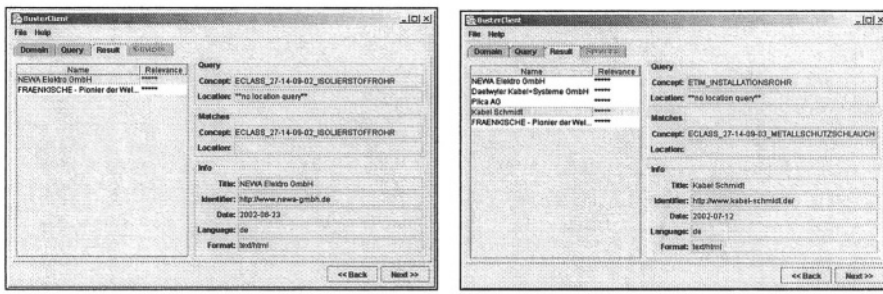
After launching the BUSTER client, the user can choose an application area (domain) (cf. figure 7.2a). Currently, we have three different domains, namely, an accommodation domain, a installation-supplies domain with parts of two well-known catalogue systems (ecl@ss and ETIM)[1], and a geographical domain (Geoshare). We choose the installation-supplies domain to demonstrate the first types of queries.

[1] www.eclass.de, www.etim.de, verified on June, 30, 2003.

Simple Concept Query

The user chooses an ontology depending on the current domain (see figure 7.2a). These terminologies are registered at the BUSTER server and are offered only when they are registered for the domain. The user is able to select one of the concepts of the ontology (eclass in this case) that fits his query best (e.g., Isolierstoffrohr (installation pipe), figure 7.2b). The BUSTER server receives the query and integrates the known ontologies for the current domain by loading them into the connected reasoner (RACER in this case)[2]. This is possible only because every ontology is annotated with a common vocabulary following the hybrid approach described in section 3.

After re-classification, all sub-concepts (children) of the query concepts form the result. Figure 7.3a shows the result of this first query. BUSTER found two annotated information sources containing the concept 'Isolierstoffrohr' (insulation pipe) from the eclass ontology (cf. query and match on the right hand side of the figure).



(a) Simple query 'Isolierstoffrohr' from the ecl@ss ontology

(b) Simple query 'Installationsrohr' from the ETIM ontology

Fig. 7.3. Result panel after querying BUSTER. In this case, a simple query has been chosen.

The power of the ontology-driven approach can be seen in figure 7.3b. We have chosen the ontology 'ETIM' and the concept 'Installationsrohr' (installation pipe). Five results are given after querying the system. We can see that, besides exact matches, semantically equivalent concepts in other ontologies are presented. Figure 7.3b reveals that the 'Installationsrohr' from the ETIM catalogue is semantically equivalent to the 'Metallschutzschlauch' (metal protection tube) from the ecl@ss ontology. Thus, this information source will be presented as an answer to the query.

[2] Usually, the user will not be asked what reasoning service should be used. However, this could make sense in certain situations, e.g., when certain features are needed (one example is concrete domains).

Defined Concept Query

Again, the user starts choosing an ontology according to the domain. Let's assume the installation-supplies ontology has been chosen. The user gets prompted with pre-defined query templates, the reason being, that common domain-dependent templates should be offered. For instance, a search for a well-known product in the installation-supply domain such as *pipes.* The user chooses a query-template provided by the BUSTER server. This template contains attributes (slots) and values (filler) from the common vocabulary. The user interface is ontology-driven, which simply means that the available attributes and fillers are dynamically loaded and presented. The user cannot make a mistake, e.g., using unknown terms. The user defines the query by selecting reasonable values for the given attributes. 'Yes' specifies the occurrence of the related filler, 'No' prohibits the occurrence and 'n/a' is chosen, if the value does not matter.

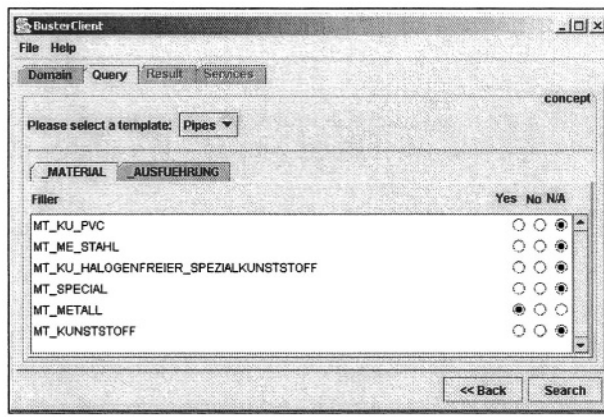
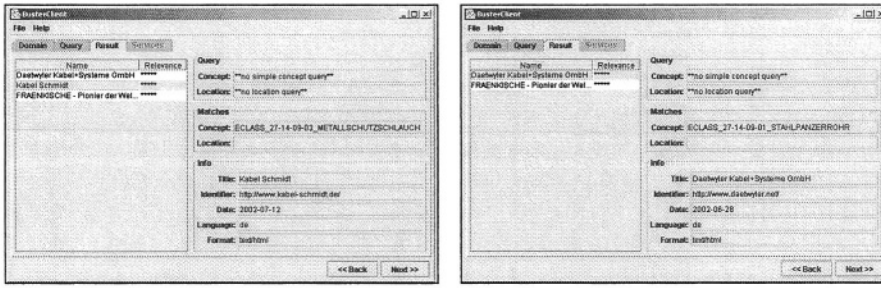


Fig. 7.4. An example for a *defined concept query:* the user has chosen the installation-supplies ontology and the query template 'pipes'.

The filled query-template is translated into a logical term. During the query process all CSDs related to the current domain are parsed for the subject-tag. Each subject is referenced to a name space, which points to an ontology that contains a concept description of the subject term. These ontologies are then downloaded from ontology servers available on the WWW, and are merged with the defined concept query and transferred into available inference machines. After re-classification, all sub-concepts (children) of the query concepts form the result.

Figure 7.4 shows an example of a defined concept query. The user is interested in information about pipe products. As we can see, the pipe template has two slots 'Material' and 'Ausführung' (material and type) along with var-



(a) Results for slot material, filler metal

(b) Results for slot material, filler metal, slot type, filler smooth

Fig. 7.5. Result panel after querying BUSTER. In this case, a defined query has been chosen (template 'pipe').

ious fillers. We set the filler 'Metall of the slot 'Material' to yes, indicating that the material of the pipe should be made out of metal.

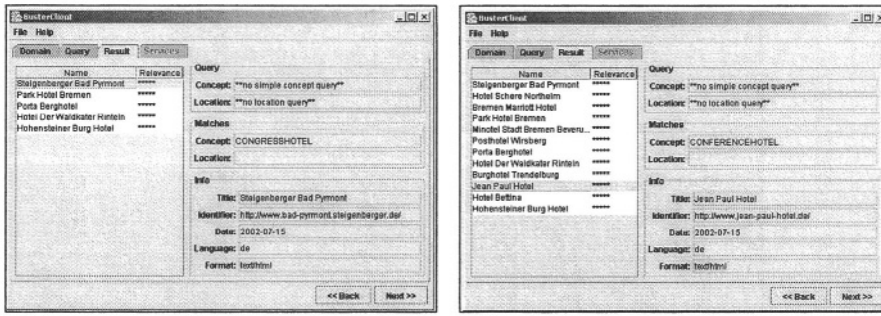
The result of the query is shown in figure 7.5a. We see that the concept "Metallschutzschlauch" of the eclass ontology fits the description made. The other results that have been found reveal that the concept "Stahlpanzerrohr" also fits the query. If the user would also fill the slot "Ausführung" (type) with "glatt" (smooth) the result would differ: figure 7.5b shows that this time, the "Metallschutzschlauch" is not found because of its rough surface.

Since RACER allows for the operation in concrete domains, BUSTER also includes templates where the user can choose this functionality. If we switch the application area from installation supplies to 'accommodation' and again choose a defined query, we would get a template allowing us to edit the three slots: 'capacity', 'single', and 'double' for an accommodation[3]. Suppose we are looking for an accommodation that includes a conference room with 80 seats. Querying the system results in five matches (7.6a). The found information sources, hotels in this case, are congress hotels. If we would change the necessary seats to, lets say 25, we would get 12 matches (7.6b).

```
(define-concept Conferencehotel
    (and
        Hotel
        (min _singles       15)
        (min _doubles       10)
        (min _capacity      30)
    )
)
```

```
(define-concept Congresshotel
    (and
        Hotel
        (min _singles       40)
        (min _doubles       20)
        (min _capacity     100)
    )
)
```

[3] At this point we have to admit that this template is somewhat awkward for the user since the intended meaning of the slots are not clear. A solution would be a proper visualization of the ontology with the concept names and attributes. This however, is hard to implement since the defined concept query does not consist of concept names but a combination of slots.



(a) Results capacity = 100 (b) Results capacity = 25

Fig. 7.6. Result panel after querying BUSTER. In this case, a defined query has been chosen, accommodation domain (template 'accommodation').

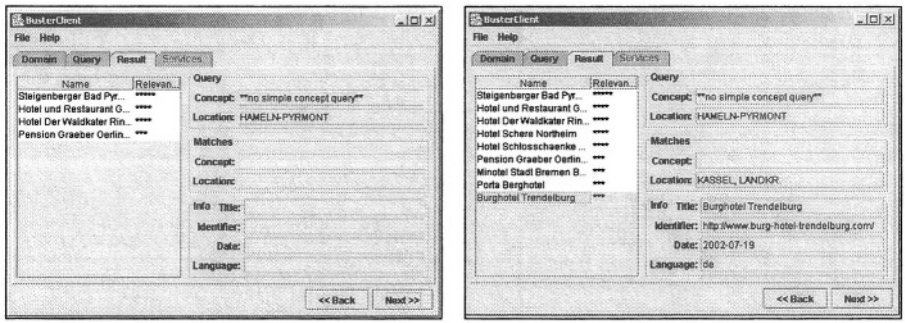
The reason for this is not that obvious. A detailed look in the accommodation T-Box[4] reveals that the concept 'congress hotel' has a minimum number of capacity, set to 100. This means that information sources (hotels) annotated with this concept provide this number of seats. This, however, also includes the requested capacity of 80. Requesting a capacity of 25 means that the minimum number of seats should be 25 which is subsumed by the concept 'congress hotel' (min 100) and the concept 'conference hotel' where the minimum is defined to be 30.

7.2.2 Spatial Queries

A user-friendly and, from a cognitive perspective, sound method to specify spatial queries as well as to index data sources and services is the use of place names as described in section 5.2. Our approach include as extension of place names, the so-called place name structures, which are based on a qualitative spatial model. These models, or spatial ontologies, use graph representations of hierarchically organized polygonal tessellations as a basis to reason about the spatial relevance of one place name with respect to another. Currently, only a few spatial models are implemented in the prototypical system[5]: a detailed qualitative model based on all German municipalities (denuts), some place name structures based on a map showing the German landscapes, place name structures for the North-Sea region, and some self-defined spatial models, e.g., hypothetic regional offices for a fictive company. The latter shows that private spatial models are allowed and can easily be integrated in the system.

In a qualitative spatial model tree, leaves corresponding to nodes of the used connection graph represent the tessellation (see figure 5.9b on page 88). Spatial relevance, a combined evaluation of partonomic and neighborhood

[4] http://www-agki.tzi.de/buster/data/ontologies/term/accommodation.racer
[5] http://www-agki.tzi.de/buster/data/ontologies/spat/



(a) Weighting factor set to neighborhood information

(b) Weighting factor set to hierarchical information

Fig. 7.7. Result panel after querying the spatial part of BUSTER.

relations between place names, is computed by calculating the horizontal and vertical (or hierarchical) graph-theoretical distances according to equation 5.10 on page 86.

The user is able to select a specific spatial ontology to initialize a spatial query. Suppose, the spatial model of Germany is selected. By selecting a place name (e.g., "Hameln-Pyrmont"), the user defines the target area of the spatial query. Using the selected spatial ontology, the spatial reasoner integrated in the BUSTER server evaluates the query and computes a list of place names that are spatially relevant to the target place name. The user is able to parameterize the query by adjusting weight sliders for horizontal and vertical relevance. The example query is configured to find only information sources that are horizontally relevant.

Figure 7.7 shows the results presented by BUSTER. Since this scenario has been described in section 5.4 on pages 87pp., a detailed explanation can be found there.

7.2.3 Temporal Queries

This part of the BUSTER system is currently under development. However, the temporal reasoning engine is already accessible by both the BUSTER server and the client. Although the system lacks comprehensive examples, one temporal model can be chosen by the user. The data we described consist of documents and information from the Bremen Senator for Construction and Environment (SBU), Referat 44. The temporal ontology contains the necessary knowledge and a reasonable differentiation for this case. Here is a part of the temporal model:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<periodNames xmlns:
    xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="http://www.tzi.de/buster/data/xsd/
                                   periodNames.xsd">
    <periodName id="Jahr_2001">
        <equals>
            <pformula name="wholeyear" year="2001" />
        </equals>
    </periodName>
    <periodName id="Jahre_1998_bis_2002">
        <begin>
            <dformula name="yearbegin" year="1998" />
        </begin>
        <end>
            <dformula name="yearend" year="2002" />
        </end>
    </periodName>
...
</periodNames>
```

The user chooses the temporal model and gets prompted with the possible templates. Suppose, he chooses the temporal concept 'Years_1998_until_2002'. The temporal reasoner expands and verifies the model as described in section 6 and calculates the temporal annotations within the CSDs of the information sources. As we can see, this temporal concept is modeled as a formula, hence, the reasoner is able to derive that a document or information source annotated with 'since_2001' fits the query. Figure 7.8 shows the result of that query.

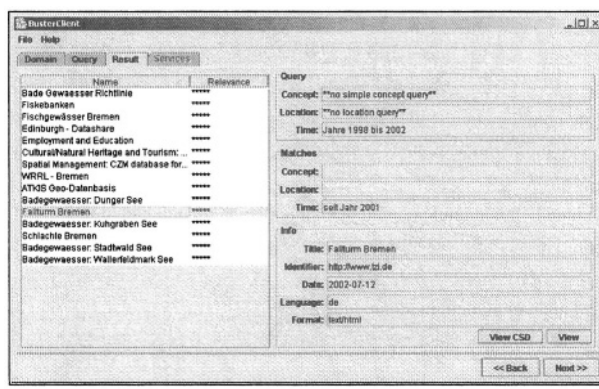


Fig. 7.8. Result panel after querying the temporal part of BUSTER.

7.3 Combined Queries

Among the described single terminological, spatial, and temporal queries, all possible combinations of queries can be made. We illustrate three additional types of queries:

- Spatio-terminological query (*concept@location*)
- Temporal-terminological query *(concept in time)*
- Spatio-temporal-terminological query (*concept@location in time*)

7.3.1 Spatio-terrninological Queries

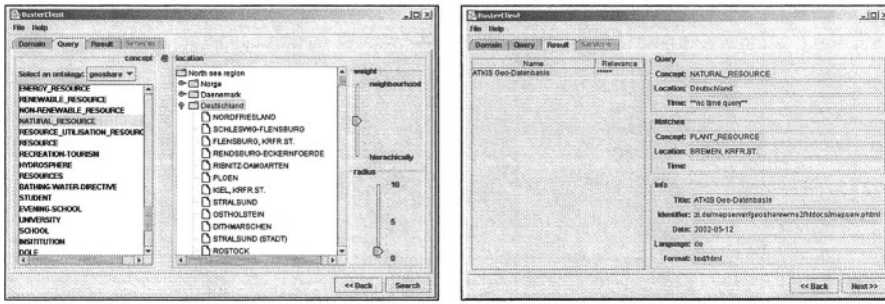
An actual ontology in the geographical domain describes terms in our European Research project 'Geoshare'. This domain contains a vast amount of concepts describing facts in the environmental area. We use an official and well-used thesaurus in this domain as a basis for the construction of the ontology called GEMET. The GEneral Multilingual Environmental Thesaurus (GEMET, [84]) was developed by the European Environment Agency (EEA) and the ETC/CDS together with a co-operation of international experts to serve the needs of environmental information systems [84]. For our demonstration, we choose this GeoShare ontology for the terminological part.

We would also like to restrict our spatial model on the self-defined spatial ontology 'North-Sea Region'. This spatial model has been automatically generated with the help of a self-developed tool called "sde2xml", which is able to transform polygons from a common GIS database into our qualitative spatial representation, the connection graph. Since the above-mentioned project is an Interreg IIIb project, we generated the qualitative spatial model for the North-Sea area. So far, the model supports Germany, The Netherlands, Denmark, Great Britain and Norway.

Suppose, we would like to find information sources that contain information about natural resources in the German area. We choose therefore the concept 'Natural_Resource' and 'Deutschland'[6]. Figure 7.9a shows the combined query.

BUSTER now combines both lists, the list of relevant concepts, and the list of spatially relevant place names, into one query. This query is applied to the BUSTER CSD database. The result is a weighted list of data sources and services matching both the terminological and the spatial query. Figure 7.9b shows the result of our combined query example. The data source found is an ATKIS Geodata data set from the SBU in Bremen which is available online. Please note that every result panel contains a service panel containing the available services (e.g., a semantic data translation service described in section 4.3).

[6] Unfortunately, the spatial models are sometimes listed in German, some minor problem that has to be fixed in the future.



(a) Combined spatio-terminological query

(b) Result of the spatio-terminological query

Fig. 7.9. Result panel after querying the BUSTER with a concept@location type

7.3.2 Temporal-Terminological Queries

This combination allows to define queries which have the type *concept in time*. This means that BUSTER is seeking information sources annotated with the given terminological concept and the given temporal concept.

Suppose we are looking for documents that contain information about natural resources (keep with our idea above) and have a certain temporal window. The query would be 'Natural_Resources' between 1998 and 2002. It is important to know that there is no spatial context and therefore, all annotations from information sources that are independent from location are calculated.

The found information source as seen in the result panel depicted in figure 7.9b would be presented again. This time, however, the location does not matter but the temporal annotation reveals that the information source is valid "since 2001". Because "since 2001" is included in the temporal concept 'Years_1998_until_2002', the information source fits the temporal query. The terminological query is the same as above and so are the results in this part.

7.3.3 Spatio-temporal-terminological Queries

The most sophisticated and interesting (from the Semantic Web point of view) type of query can be formulated as *concept@location in time*. Our example brings us in the area of tourism. We choose the application domain GeoShare for the terminological ontology, the North-Sea region as our spatial model and the temporal model from above, the SBU-Referat-44 model. Figure 7.10a shows the concepts we are looking for: we are interested in any information source or documents that contain something about fishing in the North-Sea region since 1990.

Figure 7.10b shows the result of our query. We can see that one of the found information source with the title "Fischgewässer" Bremen contains the terminological concept "angling" which is subsumed by fishing. The spatial reasoner

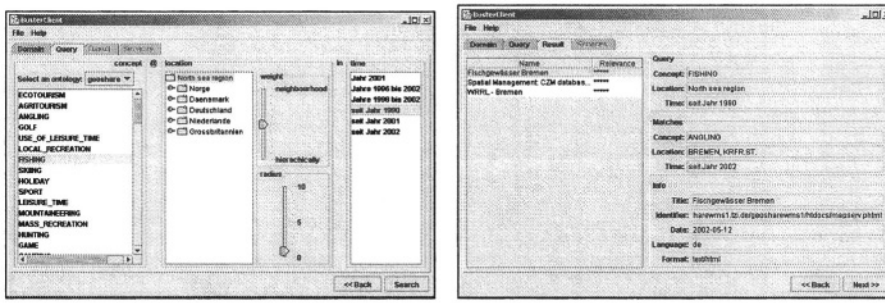


Fig. 7.10. Query and result of BUSTER with a "concept@location in time" type.

found the location "Bremen, Krfr.st." (a suburb of the city Bremen), which clearly is part of the North-Sea region and the temporal reasoner proved that the document which has been annotated with "seit Jahr 2002" also belongs to the class "seit Jahr 1990".

8

Conclusion and Future Work

We summarize the work we have done and also draw some line of research that needs to be done in the future.

8.1 Conclusion

We start with the Semantic Web sharing some ideas of what we believe is crucial for a story of success. We then discuss our BUSTER approach, following the structure of this paper and hence present three subsections discussing the results and draw conclusions.

8.1.1 Semantic Web

Whether some of the visions that be brought up in chapter 1 will become true some day is not the question. The short term visions and part of the mid-term visions are already or will become true soon. Companies are already working with or on the Semantic Web, however, in fairly limited ways. Hendler calls this “islands of the Semantic Web”. For example, one of those islands are ontologies such as the one developed at the US National Cancer Institution[1]. His vision is that those islands will be coming together over the next two years.

As described before, formal ontologies will play a major part in the Semantic Web. One question that arises is: which kind of language will be the “official” ontology language? This is not foreseeable right now, however, it looks like OWL could play this role. Our opinion is, that the major problems with regard to expressiveness etc. are more or less solved. There will be some minor corrections in the future, the major subject although is that the people involved in those working groups come together and finally agree on some standard.

[1] http://www.mindswap.org/2003/CancerOntology, verified on June 15, 2003.

One important aspect is the description of information. This is a crucial part since the information is the reason why we use the Web. Metadata need to be acquired automatically as much as possible so that the "real" information can be annotated properly. We also believe that more tools are needed providing the ordinary user of the Web with help to annotate their data.

8.1.2 BUSTER Approach and System

The most important result of our work is that our approach, both the conceptual and the implementation part, is operating the way we wanted it to operate. This includes all the requirements that have been defined before we started the work.

An important result is the type of queries that are possible. We are able to support the user (or other systems) with new types of queries because of the development of the spatial and temporal reasoners. These queries are *concept@location, concept in time,* or *concept@location in time.* This types of queries can help to support users or systems in finding what they are after in a more intelligent and accurate manner.

Another major result is the improvement of expressiveness. We called the requirement "intuitive labeling" (e.g., place names, period names) and implemented this throughout our system. This is an important part of our approach enabling users to use colloquial terms while editing their search.

A new service, which we call semantic translation, will be enabled automatically if the necessary contexts and the required ontologies are existing. This service (which is by the way not a Web service) is able to transform information on the data level from one context to another. We might add and emphasize that this is a major difference to information integration on the concept level.

The BUSTER system is currently being used within two research projects. The BMBF (German Ministry for Education and Research) project meanInGS[2] deals with semantic interoperability problems and Geo-Services (in a Web service sense). We use our approach to seek geo-objects and for the mapping between catalogues. We also work on Web services that can be chained in order to provide users with better answers. The second project GeoShare[3], funded by the EU, deals with the development of user centric services to support better governance, democratic processes and a sustainable and balanced development of rural and urban areas around the North Sea.

Terminological Part

The most important result in this part is that the representation and reasoning with the help of description-logics-based approaches is sufficient enough to

[2] http://www.meanings.de
[3] http://www.geoshare.net

meet the given requirements. We do not want to be in favor for a specific language because a number of languages do support what we need. However, one demand is a proper support by reasoning engines which is provided by only a few approaches.

Another major outcome is the approach of using a hybrid ontology approach. This means to have multiple ontologies (usually one for each source) that use the same common vocabulary. Our opinion is that this approach can be used at least within one community. People involved in the current research projects meanInGS and GeoShare confirm this position.

One necessary element to describe the content of data or information sources are metadata. A thorough study revealed that some existing metadata standards can be adopted to meet our requirements [132]. We have chosen the Dublin Core standard and developed new qualifiers for our purpose. We call this the comprehensive source description and it turned out that the concept works well.

Spatial Part

We have shown that our approach meets the requirements that we think are necessary to support both annotation and intelligent retrieval of spatial data. Our most important requirement, the intuitive labeling of geographic regions/places, can be fulfilled using our place names or place name structure approach.

The new footprint based on a standard reference tessellation gives us the option to map arbitrary place name structures onto common reference units such as zip codes or administrative units.

Our new reasoning approach based on connection graphs is able to perform inferences for a new type of reasoning: spatial relevance reasoning. Whether a polygon is spatially relevant can be determined by means of a combination of neighborhood information and partonomic information.

So far, we deal with static spatial knowledge. The idea of including changing spatial knowledge has also to be considered in the future.

Temporal Part

We showed that the existing temporal approaches are not satisfactory to serve the requirements of the modern Semantic Web. The major problem is the lack of expressiveness and the non-existing solutions for intuitive labeling and annotation of data sources.

We developed a new representation scheme allowing us to define exact, fuzzy, persistent, and unknown boundaries. In addition, we are able to define internal relations or referrals which means that we can define a boundary of an interval with the help of a reference to the boundary of another interval. This leads to quite a number of possible combinations, which are supported as well.

Our developed and implemented temporal reasoning engine supports these requirements. The engine is a powerful tool to both check the underlying temporal model for consistency and derive new information hidden in the model. We think that this is an important step forward in the area of temporal annotation and reasoning with regard to the Semantic Web.

8.2 Future Work

The work that we have done so far can be extended in almost every part. Right now, we would like to discuss the major improvements that can be made.

8.2.1 Terminological Part

A major drawback using some kind of description logics is the fact that a concept is either subsumed by another concept or it is not. This black/white paradigm is something that does not fit well to reality. One idea, that is already followed by some researchers [109], is known as “approximate terminological reasoning”.

The crucial part is the annotation of information sources based on the ontologies used. We believe that there is a need for automatic annotation tools to support the user with this work. First ideas have already been published [79], however, more work has to be done in this area.

8.2.2 Spatial Part

Further developments will include the ability to add new place names in already existing place name structures. This also includes the *extensionalization* of intensionally defined place names automatically. This means that an added place name can be mapped to the underlying reference units automatically. Usually, the knowledge engineer has to take care of this step. The first developments in this direction are described in Vögele et al. [120].

8.2.3 Temporal Part

Future research concentrates on more relations that have to be integrated in the reasoning engine. We will also offer a small temporal reasoning service on the Web, which everybody is able to access to.

Another important step is to add more temporal relations and relax the restriction to *older, younger, contemporary, survives* and *survived-by.* A proper way to a solution would be using the conceptual neighborhoods *head-to-head* and *tail-to-tail* relations to declare the simultaneous beginning or end of time intervals.

References

1. AdV. *Amtliches Topographisch-Kartographisches Informationssystem ATKIS.* Landesvermessungsamt NRW, Bonn, 1998.
2. J. F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM,* 26(11):832–843, 1984.
3. Yigal Arens, Chun-Nan Hsu, and Craig A. Knoblock. Query processing in the SIMS information mediator. In Michael N. Huhns and Munindar P. Singh, editors, *Readings in Agents,* pages 82–90. Morgan Kaufmann, San Francisco, CA, USA, 1997.
4. F. Baader, H.-J. Heinsohn, B. Hollunder, J. Müller, B. Nebel, W. Nutt, and H.-J. Profitlich. Terminological knowledge representation: A proposal for a terminological logic. Technical memo tm-90-04, Deutsches Forschungszentrum für Künstliche Intelligenz GmbH (DFKI), 1991.
5. Peter van Beek. Reasoning about qualitative temporal information. *Artificial Intelligence,* 58:297–326, 1992.
6. B. Bennett. Spatial reasoning with propositional logics. In J. Doyle, E. Sandewall, and P. Torasso, editors, *Principles of Knowledge Representation and Reasoning* (*KR 94*), San Francisco, CA, USA, 1994. Morgan Kaufman.
7. Tim Berners-Lee, Dan Brickley, Dan Connolly, Mike Dean, Stefan Decker, Dieter Fensel, Richard Fikes, Pat Hayes, Jeff Heflin, Jim Hendler, Ora Lassila, Deb McGuinness, and Lynn Andrea Stein. Daml+oil language specification, 2001. Web page: http://www.daml.org/2001/03/daml+oil-index.html, verified on July, 1st, 2003.
8. Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific American,* 2001(5), 2001.
9. Grady Booch, James Rumbaugh, and Ivar Jacobson. *The unified modeling language user guide.* The Addison-Wesley object technology series. Addison-Wesley, Reading Mass., 1999.
10. A. Borgida, R. J. Brachman, D. L. McGuiness, and L. A. Resnick. Classic: A structural data model for objects. In *ACM SIGMOID International Conference on Management of Data,* Portland, Oregon, USA, 1989.
11. T. Bray, J. Paoli, and C.M. Sperberg-McQueen. Extensible markup language (xml) 1.0 (second edition). Technical Report REC-xml-20001006, W3C, 2000 2000. http://www.w3.org/TR/2000/REC-xml-20001006 http://www.w3.org/TR/REC-xml.

12. Diego Calvanese, Giuseppe De Giacomo, and Maurizio Lenzerini. Description logics for information integration. In A. Kakas and F. Sadri, editors, *Computational Logic: Logic Programming and Beyond, Essays in Honour of Robert A. Kowalski,* volume 2408 of *Lecture Notes in Computer Science,* pages 41–60. Springer, 2002.
13. Diego Calvanese, Giuseppe De Giacomo, and Maurizio Lenzerini. Description logics for information integration. In *Computational Logic: From Logic Programming into the Future (In honour of Bob Kowalski*), Lecture Notes in Computer Science. Springer-Verlag, 2001. To appear.
14. S. Chawathe, H. Garcia-Molina, J. Hammer, K. Ireland, et al. The tsimmis project: Integration of heterogeneous information sources. In *Conference of the Information Processing Society Japan,* pages 7–18, 1994.
15. W. W. Cohen, A. Borgida, and H. Hirsh. Computing least common subsumers in description logics. In *AAAI - 92,* pages 754–760. AAAI Press/The MIT Press, 1992.
16. Anthony Conn. Qualitative spatial representation and reasoning techniques. In *KI-97: Advances in Artificial Intelligence,* pages 1–30, Berlin, 1997. Springer.
17. Anthony Cohn and Shyamanta Hazarika. Qualitative spatial representation and reasoning: An overview. *Fundamenta Informaticae,* 46(1-2):1–29, 2001.
18. Christine Collet, Michael N. Huhns, and Wei-Min Shen. Resource integration using a large knowledge base in carnot. *IEEE Computer,* 24(12):55–62, 1991. Describes Project CARNOT: CYC as global schema for local information sources. Mappings between the local and global scheme is be done due to "articulation axioms" (= IST in context logic). This Paper introduced in CARNOT and gives guidelines how to build the articulation axioms.
19. Isabel Cruz, Stefan Decker, Jèrôme Euzenat, and Deborah McGuinness, editors. *Proceedings of the 1st Semantic Web Working Symposium* (*SWWS*). Stanford University, 2001.
http://www.semanticweb.org/ SWWS/program/full/SWWSProceedings.pdf, verified on May, 15, 2003.
20. DCMI. Dublin core metadata element set, version 1.1: Reference description, 1999. http://dublincore.org/documents/1999/07/02/dces/#rfc2413.
21. DCMI Usage Board. Dcmi type vocabulary, 2003. Report on web page:
http://dublincore.org/documents/dcmi-type-vocabulary/,
verified on July, 1st, 2003.
22. T. de Laguna. Point, line and surface as sets of solids. *The Journal of Philosophy,* 19:449–461, 1922.
23. Rina Dechter, Itay Meiri, and Judea Pearl. Temporal constraint networks. *Artificial Intelligence,* 49:61–95, 1991.
24. Edsger W. Dijkstra. A note on two problems in connexion with graphs. *Numeriskche Mathematik,* 1:269–271, 1959.
25. F. Donini, M. Lenzerini, D. Nardi, and A. Schaerf. Al-log: Integrating datalog and description logics. *Journal of Intelligent Information Systems* (*JIIS*), 27(1), 1998.
26. Max Egenhofer. Reasoning about binary topological relations. In O. Günther and H.-J. Schek, editors, *Symposium on Large Spatial Databases SSD,* volume 525 of *Lecture Notes in Computer Science,* pages 143–160. Springer, 1991.
27. Max Egenhofer and R. Frenzosa. Point-set topological spatial relations. *International Journal of Geographic Information Systems,* 5(2):161–174, 1991.

28. M. T. Escrig and F. Toledo. Qualitative spatial reasoning: Theory and practice - application to robot navigation. In *Frontiers in AI and Applications,* volume 47. IOS Press, 1998.
29. ETC-CDS. General european multilingual environement thesaurus (gemet), 17.03.2000 2000. European Topic Centre on Catalogue of Data Sources.
30. European Environmental Agency. Corine land cover. Technical guide, Commission of the European Communities OPOCE (Office for official publications of the european communities) ©ECSC-EEC-EAEC, 1997-1999 1991. http://reports.eea.eu.int/COR0-landcover/en.
31. D. Fensel, I. Horrocks, F. Van Harmelen, S. Decker, M. Erdmann, and M. Klein. Oil in a nutshell. In *12th International Conference on Knowledge Engineering and Knowledge Management EKAW 2000,* Juan-les-Pins, France, 2000.
32. Dieter Fensel, Jörg Angele, Stefan Decker, Michael Erdmann, Hans-Peter Schnur, Steffen Staab, Rudi Studer, and A. Witt. On2broker: Semantic-based access to information sources at the www. In P.D. Bra and J. J. Legget, editors, *World Conference on the WWW and Internet* (*WebNet*), pages 366–371, Charlottesville, VA, USA, 1999. Association for the Advancement of Computing in education (AACE).
33. Dieter Fensel, Frank van Harmelen, Ian Horrocks, Deborah L. McGuinness, and Peter F. Patel-Schneider. Oil: An ontology infrastructure for the semantic web. *IEEE Intelligent Systems,* 16(2):38–44, 2001.
34. K. Forbus, P. Nielson, and B. Faltings. Qualitative kinematics: A framework. In *International Joint Conference on Artificial Intelligence* (*IJCAI*), pages 430–436, Milan, Italy, 1987. Morgan Kaufman.
35. Christian Freksa. Temporal reasoning based on semi-intervals. *Artificial Intelligence,* 54(1):199–227, 1992.
36. Anthony Galton. The mereotopology of discrete space. In Christian Freksa and David Mark, editors, *Spatial Information Theory - Cognitive and Computational Foundations of Geographic Information Science* (*COSIT*), volume 1661 of *Lecture Notes in Computer Science* (*LNCS*), pages 251–266, Stade, Germany, 1999. Springer Verlag.
37. Rosella Gennari. *Temporal Reasoning and Constraint Programming: A Survey.* Masters thesis, Universiteit van Amsterdam, 1998.
38. G. Gerla. Pointless geometries. In F. Buekenhout, editor, *Handbook of Incidence Geometry,* pages 1015–1031. Elsevier Science bv, 1995.
39. Gabriele Giesenberg. Kirchliches und städtisches leben. In Uta von Freeden and Siegmar von Schnurbein, editors, *Spuren der Jahrtausende - Archäologie und Geschichte in Deutschland,* pages 390–417. Konrad Theiss Verlag Stuttgart, Frankfurt am Main, 2002.
40. François Goasdoué, Véronique Lattes, and Marie-Christine Rousset. The use of carin language and algorithms for information integration: The picsel project,. *International Journal of Cooperative Information Systems* (*IJCIS*), 9(4):383 – 401, 1999.
41. Cheng Hian Goh. *Representing and Reasoning about Semantic Conflicts in Heterogeneous Information Sources.* Phd, MIT, 1997.
42. N. M. Gotts, J. Gooday, and Anthony Cohn. A connection-based approach to common sense topological description and reasoning. *The Monist,* 79(1):51–75, 1996.

43. Michael Grüninger and Mike Uschold. Ontologies and semantic integration, 2002. 1. Government report on the state of the art and future predictions for agent technology.
44. Thomas R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition,* 5(2): 199–220, 1993.
45. Thomas R. Gruber. Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human Computer Studies,* 43(5/6):907–928, 1995.
46. Volker Haarslev and Ralf Möller. Spatioterminological reasoning: Subsumption based on geometric reasoning. In Rousset et al., editor, *11th International Workshop on Description Logics DL'97,* pages 74–76, Gif sur Yvette, France, 1997. Universite Paris Sud, Laboratoire de Recherche en Informatique (LRI).
47. Volker Haarslev and Ralf Möller. High performance reasoning with very large knowledge bases. In Bernhard Nebel, editor, *International Joint Conferences on Artificial Intelligence (IJCAI),* volume 1, pages 161–166, Seattle, WA, 2001. Morgan Kaufman.
48. Frank van Harmelen, Jim Hendler, Ian Horrocks, Deborah L. McGuinness, Peter F. Patel-Schneider, and Lynn Andrea Stein. Owl web ontology language reference, http://www.w3.org/tr/owl-ref/, verified on july, 1st, 2003, 31 March 2003 2003. Web page: http://www.w3.org/TR/owl-ref/.
49. Andy Harter, Andy Hopper, Pete Steggles, Andy Ward, and Paul Webster. The anatomy of a context-aware application. In *Mobile Computing and Networking,* pages 59–68, 1999.
50. Patrick Hayes. A catalog of temporal theories. Technical report UIUC-BI-AI-96-01, University of Illinois 1995, 1996.
51. Patrick Hayes. Semantic web (interview). *Künstliche Intelligenz,* 03/2003:41–42, 2003.
52. Sebastian Hübner. *Qualitative Abstraktion von Zeit für Annotation und Retrieval im Semantic Web.* Mastersthesis, Universität Bremen, 2003.
53. Jeff Heflin and James Hendler. A portrait of the semantic web in action. *IEEE Intelligent Systems,* 16(2):54–59, 2001.
54. James Hendler. Semantic web (interview). *Künstliche Intelligenz,* 03/2003:39–40, 2003.
55. D. Hernández. Qualitative representation of spatial knowledge. In *Lecture Notes in Artificial Intelligence,* volume 804. Springer, 1994.
56. Linda Hill. Adl gazetteer content standard, 2000. http://www.alexandria.ucsb.edu/gazetteer/gaz_content_standard.html.
57. Linda Hill. Core elements of digital gazetteers: placenames, categories, and footprints. In José Luis Borbinha and Thomas Baker, editors, *ECDL,* volume 1923 of *Lecture Notes in Computer Science,* pages 280–290. Springer, 2000.
58. S. C. Hirtle and J. Jonides. Evidence of hierarchies in cognitive maps. *Memory & Cognition,* 13:208–217, 1985.
59. I. Horrocks. Fact and ifact. In P. Lambrix, A. Borgida, M. Lenzerini, R. Möller, and P. Patel-Schneider, editors, *Proceedings of the International Workshop on Description Logics* (*DL'99*), pages 133–135. CEUR-Workshop Proceedings at http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS, 1999.
60. Ian Horrocks and James Hendler, editors. *The Semantic Web - ISWC 2002, First International Semantic Web Conference, Sardinia, Italy, June 9-12, 2002, Proceedings,* volume 2342 of *Series: Lecture Notes in Computer Science.* Springer-Verlag, 2002.

61. Ian Horrocks, Ulrike Sattler, and Stephan Tobies. Practical reasoning for very expressive description logics. *Logic Journal of the IGPL,* 8(3):239–263, 2000.
62. Ian Horrocks, Ulrike Sattler, and Stephan Tobies. Reasoning with individuals for the description logic SHIQ. In David MacAllester, editor, *Proceedings of the 17th International Conference on Automated Deduction* (*CADE-17*), number 1831 in LNAI, pages 482–496, Germany, 2000. Springer Verlag.
63. Dublin Core Metadata Initiative. The dublin core: A simple content description model for electronic resources, 28.03.2000 2000. http://purl.org/dc.
64. V. Kashyap and A. Sheth. Schematic and semantic similarities between database objects: A context-based approach. *The International Journal on Very Large Data Bases,* 5(4):276–304, 1996.
65. Vipul Kashyap and Amit Sheth. Semantic heterogeneity in global information systems: The role of metadata, context and ontologies. In Michael P. Papazoglou and Gunter Schlageter, editors, *Cooperative Information Systems,* pages 139–178. Academic Press, San Diego, 1998.
66. Vipul Kashyap and Amit Sheth. Semantic heterogeneity in global information systems: The role of metadata, context and ontologies. In M. Papazoglou and G. Schlageter, editors, *Cooperative Information Systems: Current Trends and Applications,* pages 139–178. Academic Press, San Diego, 1998.
67. Won Kim, Injun Choi, Sunit Gala, and Mark Scheevel. On resolving schematic heterogeneity in multidatabase systems. In Won Kim, editor, *Modern Database Systems: The Object Model, Interoperability, and Beyond,* pages 521–550. ACM Press/Addison-Wesley Publishing Company, 1995.
68. Won Kim and Jungyun Seo. Classifying schematic and data heterogeinity in multidatabase systems. *IEEE Computer,* 24(12):12–18, 1991. problem classification of semantic heterogeneity.
69. Michel Klein. Combing and relating ontologies: an analysis of problems and solutions. In A. Gomez-Perez, M. Gruninger, H. Stuckenschmidt, and M. Uschold, editors, *IJCAI-01,* volume 47, Seattle, WA, 2001. CEUR.
70. Michel Klein. Xml, rdf, and relatives. *IEEE Intelligent Systems,* 16(2):28–28, 2001.
71. C. A. Knoblock, S. Minton, J. L. Ambite, N. Ashish, I. Muslea, A. G. Philpot, and S. Tejada. The ariadne approach to web-based information integration. *International Journal of Cooperative Information Systems* (*IJCIS*), 10(1-2):145–169, 2001.
72. Alon Y. Levy and Marie-Christine Rousset. Carin: A representation language combining horn rules and description logics. In *Proceedings of the 12th European Conf. on Artificial Intelligence* (*ECAI-96*), pages 323–327, 1996.
73. Alon Y. Levy, Divesh Srivastava, and Thomas Kirk. Data model and query evaluation in global information systems. *Journal of Intelligent Information Systems* (*JIIS*), 5(2):121–143, 1995.
74. Gérard Ligozat. A new proof of tractability for ord-horn relations. In *AAAI 96: 13th National Conference on Artificial Intelligence. IAAI 96: 8th Conference on Innovative Applications of Artificial Intelligence.,* pages 395–401, Portland, Oregon, 1996. AAAI-Press.
75. Gérard Ligozat. "corner" relations in allen's algebra. *Constraints,* 3(2/3):165–177, 1998.
76. William Bryant Logan, Vance Muse, Donald Young, and Roger G. Kennedy. *The Deep South (Smithsonian Guides to Historic America).* Stewart, Tabori & Chang, revised edition, 1998.

77. Marwa Mabrouk, Harry Niedzwiadek, Yaser Bishr, Jonathan Williams, et al. Xml for location services (xls): The openls platform. Discussion paper and recommendation OGC 02-211, Version 0.2.0, Open GIS Consortium Inc., 16. Dezember 2002 2002.
78. Robert M. MacGregor. Using a description classifier to enhance deductive inference. In *Proceedings Seventh IEEE Conference on AI Applications,* pages 141–147, 1991.
79. Alexander Maedche and Steffen Staab. Ontology learning for the semantic web. *IEEE Intelligent Systems,* 16(2):72–79, 2001.
80. C. Masolo and L. Vieu. Atomicity vs. infinite divisibility of space. In Christian Freksa and David Mark, editors, *Spatial Information Theory - Cognitive and Computational Foundations of Geographic Information Science (COSIT),* volume 1661 of *Lecture Notes in Computer Science (LNCS),* pages 235–250, Stade, Germany, 1999. Springer Verlag.
81. Eduardo Mena, Vipul Kashyap, Amit P. Sheth, and Arantza Illarramendi. OBSERVER: An approach for query processing in global information systems based on interoperation across pre-existing ontologies. In *Conference on Cooperative Information Systems,* pages 14–25, 1996.
82. R. Möller, V. Haarslev, and B. Neumann. Semantics-based information retrieval. In J. Cuena, editor, *IT & KNOWS Information Technology and Knowledge Systems, XV. IFIP World Computer Congress,* Vienna, Budapest, 1998. Austrian Computer Society.
83. C. F. Naiman and A. M. Ouksel. A classification of semantic conflicts in heterogeneous database systems. *Journal of Organizational Computing,* pages 167–193, 1995.
84. Kirsten Nax and Justina Lethen. Etc/cds general multilingual environmental thesaurus (gemet) - general information on gemet. Technical report, ETC/CDS, 1999.
85. Bernhard Nebel. Artificial intelligence: A computational perspective. In G. Brewka, editor, *Principles of Knowledge Representation,* Studies in Logic, Language and Information, pages 237–266. CSLI publications, Stanford, 1996.
86. Bernhard Nebel and Hans-Jürgen Bürckert. Reasoning about temporal relations: a maximal tractable subclass of allen's interval algebra. *Journal of the ACM,* 42(1):43–66, 1995.
87. Holger Neumann, Gerhard Schuster, Heiner Stuckenschmidt, Ubbo Visser, and Thomas Vögele. Intelligent brokering of environmental information with the buster system. In Lorenz M. Hilty and Paul W. Gilgen, editors, *International Symposium Informatics for Environmental Protection,* volume 30 of *Umwelt-Informatik Aktuell,* pages 505–512, Zürich, Switzerland, 2001. Metropolis.
88. OGC. The opengisŹ abstract specification - topic 14: Semantics and information communities. OpenGIS Project Document Number 99-114, OpenGIS Consortium, 1999.
89. Yannis Papakonstantinou, Hector Garcia-Molina, and Jeffrey Ullman. Medmaker: A mediation system based on declarative specifications. In *International Conference on Data Engineering,* pages 132–141, New Orleans, 1996.
90. Terry Payne and Eric Miller. Calendars, schedules and the semantic web. *European Research Consortium for Informatics and Mathematics,* 51(October):16–17, 2002. http://www.ilrt.bris.ac.uk/discovery/2001/04/calendar/.
91. Ernst Pitz. Mittelalter. In *Lexikon des Mittelalters,* volume 6, pages 684–687. Deutscher Taschenbuch Verlag, München, 2002.

92. Alun D. Preece, Kit ying Hui, W. A. Gray, P. Marti, Trevor J. M. Bench-Capon, D. M. Jones, and Zhan Cui. The KRAFT architecture for knowledge fusion and transformation. *Knowledge Based Systems,* 13(2-3): 113–120, 2000.
93. David A. Randell, Zhan Cui, and Anthony G. Cohn. A spatial logic based on regions and connection. In B. Nebel, W. Swartout, and C. Rich, editors, *Knowledge Representation and Reasoning KRR,* pages 165–176, Cambridge, 1992. Morgan Kaufman.
94. A.L. Rector, S. Bechofer, C.A. Goble, I. Horrocks, W.A. Nowlan, and W.D. Solomon. The grail concept modelling language for medical terminology. *Artificial Intelligence in Medicine,* 9:139 – 171, 1997.
95. R. Röhrig. A theory of qualitative spatial reasoning based on order relations. In *AAAI 94,* pages 1418–14232. AAAI Press, 1994.
96. Wolf-Fritz Rieckert. Erschlieîung von fachinformationen im internet mit hilfe von thesauri und gazetteers. In Christian Dade and Bernhard Schulz, editors, *Management von Umweltinformationen in vernetzten Umgebungen,* volume 21 of *Reihe Umweltinformatik aktuell,* page 240. Metropolis, Nürnberg, 1999.
97. Safe Software Inc. Semantic data translation using fme. White paper, URI: http://www.safe.com/solutions/whitepapers/semantic_data_translation.htm, 2003.
98. Safe Software Inc. Semantic translation. White paper, URI: http://www.safe.com/solutions/whitepapers/semantic_translation.htm, 2003.
99. Christoph Schlieder, Thomas Vögele, and Ubbo Visser. Qualitative spatial representation for information retrieval by gazetteers. In *Conference of Spatial Information Theory COSIT,* volume 2205 of *Spatial Information Theory: Foundations of Geographic Information Science,* pages 336–351, Morrow Bay, CA, 2001. Springer.
100. Manfred Schmidt-Schauî and Gert Smolka. Attributive concept descriptions with complements. *Artificial Intelligence,* 48(1):1–26, 1991.
101. Eddie Schwalb and Rina Dechter. Coping with disjunctions in temporal constraint satisfaction problems. In *The National Conference on Artificial Intelligence (AAAI-93),* pages 127–132, Washington, D.C., July, 1993. AAAI Press.
102. Eddie Schwalb and LLuís Vila. Temporal constraints: A survey. *Constraints,* 3(2/3):129–149, 1998.
103. Ifan Shepherd. Information integration in gis. In D. J. Maguire, M. F. Goodchild, and D. W. Rhind, editors, *Geographical Information Systems: Principles and applications,* volume 1, pages 337–360. Longman, London, UK, 1991.
104. Barry Smith and David M. Mark. Ontology and geographic kinds. In T. K. Poiker and N. Chrisman, editors, *8th International Symposium on Spatial Data Handling (SDH'98),* pages 308–320, Vancouver, Canada, 1998. International Geographical Union.
105. Mark Stefik. *Knowledge Systems.* Morgan Kaufman Publishers Inc., San Francisco, CA, 1995.
106. R. Stevens, P. Baker, S. Bechhofer, G. Ng, A. Jacoby, N.W. Paton, C.A. Goble, and A. Brass. Tambis: Transparent access to multiple bioinformatics information sources. *Bioinformatics,* 16(2):184–186, 2000.
107. Oliviero Stock, editor. *Spatial and Temporal Reasoning.* Kluwer Academic Publishers, Dordrecht, NL, 1997.

108. H. Stuckenschmidt, Frank van Harmelen, Dieter Fensel, Michel Klein, and Ian Horrocks. Catalogue integration: A case study in ontology-based semantic translation. Technical Report IR-474, Computer Science Department, Vrije Universiteit Amsterdam, 2000.
109. Heiner Stuckenschmidt. Approximate information filtering with multiple classification hierarchies. *International Journal on Computational Intelligence and Applications, Special Issue on Intelligent Web Applications,* 2(3):295–302, 2003.
110. Heiner Stuckenschmidt, Thomas Vögele, Ubbo Visser, and Ryco Meyer. Intelligent brokering of environmental information with the buster system. In *In: Information Age Economy: Proceedings of the 5th International Conference 'Wirtschaftsinformatik',* pages 15–20, Ulm, Germany, 2001. Physica-Verlag.
111. Heiner Stuckenschmidt and Ubbo Visser. Semantic translation based on approximate re-classification. In *Workshop on Semantic Approximation, Granularity and Vagueness, Workshop of the Seventh International Conference on Principles of Knowledge Representation and Reasoning,* pages 110–118, Breckenridge, 2000.
112. Heiner Stuckenschmidt, Ubbo Visser, Christoph Schlieder, Thomas Vögele, and Holger Neumann. Spatial reasoning for information brokering. In Ingrid Russell and John Kolen, editors, *Florida Artificial Intelligence Research Society Conference (FLAIRS),* pages 568 – 573, Key West, FL, 2001. AAAI Press.
113. Heiner Stuckenschmidt, Ubbo Visser, Gerhard Schuster, and Thomas Vögele. Ontologies for geographic information integration. In Ubbo Visser and Hardy Pundt, editors, *Workshop "Intelligent Methods in Environmental Protection: Special Aspects of Processing in Space and Time", 13. International Symposium of Computer Science for Environmental Protection (CSEP '99),* volume 5 of *Research reports of the Department of Mathematics and Computer Science, University of Bremen,* pages 81–107. University of Bremen, 1999. sw.
114. Heiner Stuckenschmidt and Holger Wache. Context modelling and transformation for semantic interoperability. In Mokrane Bouzeghoub, Matthias Klusch, Werner Nutt, and Ulrike Sattler, editors, *Knowledge Representation Meets Databases (KRDB 2000),* volume 29, page 14. CEUR Workshop Proceedings, 2000.
115. The Unicode Consortium. The Unicode standard, version 2.0. Technical paper, Addison-Wesley Developers Press, 1996.
116. Sabine Timpf. Ontologies of wayfinding. *Networks and Spatial Economics,* 2(1):9–33, 2002.
117. Waldo Tobler. A computer movie simulating urban growth in the detroit region. *Economic Geography,* 46(2):234–240, 1970. The Śfirst law of geographyŠ is at the bottom of page 236.
118. Andrej Vckovski. *Interoperable and Distributed Processing in GIS.* Taylor & Francis, London, 1998.
119. Thomas Vögele and Christoph Schlieder. Spatially-aware information retrieval with place names. In *16th international FLAIRS conference,* St. Augustine, FL, USA, 2003. AAAI Press. to appear.
120. Thomas Vögele, Christoph Schlieder, and Ubbo Visser. Intuitive modelling of place name regions for spatial information retrieval. In *COSIT,* Ittingen, Switzerland, 2003. Springer. to appear.
121. Thomas Vögele, Heiner Stuckenschmidt, and Ubbo Visser. Towards intelligent brokering of geoinformation. In *Urban and Rural Data Management (UDMS),* pages 135–142, Delft, 2000. Electronic. sw.

122. L. Vieu. Spatial representation and reasoning in ai. In O. Stock, editor, *Spatial and Temporal Reasoning,* pages 3–40. Kluwer, 1997.
123. LLuís Vila. A survey on temporal reasoning in artificial. *AI Communications,* 7(1):4–28, 1994.
124. Marc B. Vilain and Henry A. Kautz. Constraint propagation algorithms for temporal reasoning. In Tom Kehler and Stan Rosenschein, editors, *AAAI,* pages 377–382, Philadelphia, PA, 1986. AAAI Press.
125. Ubbo Visser and Sebastian Hübner. Temporal representation and reasoning for the semantic web. Technical TZI-Bericht Br. 28, 2003, TZI, Center for Computing Technologies, 2003.
126. Ubbo Visser and Christoph Schlieder. Modeling real estate transactions: the potential role of ontologies. In Heiner Stuckenschmidt, Erik Stubkjaer, and Christoph Schlieder, editors, *The Ontology and Modelling of Real Property Transactions,* International Land Management Series, pages 115–130. Ashgate Publishing Limited, ISBN 0 7546 3287 3, Aldershot, Hants GU11 3HR, 2002.
127. Ubbo Visser and Gerhard Schuster. Finding and integration of information - a practical solution for the semantic web -. In Jérôme Euzénat, Asuncion Gomez-Perez, Nicola Guarino, and Heiner Stuckenschmidt, editors, *ECAI 02, Workshop on Ontologies and Semantic Interoperability,* pages 73–78, Lyon, France, 2002. ECCAI.
128. Ubbo Visser and Heiner Stuckenschmidt. Intelligent location-dependent acquisition and retrieval of environmental information. In *21st Urban Data Management Symposium,* pages 130–138, Vienna, Italy, 1999. The Urban Data Management Society. sw.
129. Ubbo Visser, Heiner Stuckenschmidt, and Christoph Schlieder. Interoperability in gis - enabling technologies. In Maurici Ruiz, Michael Gould, and Jer=nica Ramon, editors, *5th AGILE Conference on Geographic Information Science,* pages 291–297, Palma de Mallorca, Spain, 2002. Universitat de les Illes Balears.
130. Ubbo Visser, Heiner Stuckenschmidt, Christoph Schlieder, Holger Wache, and Ingo Timm. Terminology integration for the management of distributed information resources. *Künstliche Intelligenz (KI), Special Issue Knowledge Management,* 16(1):31–34, 2002.
131. Ubbo Visser, Heiner Stuckenschmidt, Gerhard Schuster, and Thomas Vögele. Ontologies for geographic information processing. *Computers & Geosciences,* 28(1):103–118, 2002.
132. Ubbo Visser, Heiner Stuckenschmidt, Holger Wache, and Thomas Vögele. Using environmental information efficiently: Sharing data and knowledge from heterogeneous sources. In Claus Rautenstrauch and Susanne Patig, editors, *Environmental Information Systems in Industry and Public Administration,* pages 41–73. IDEA Group, Hershey, USA & London, UK, 2001.
133. Ubbo Visser, Thomas Vögele, and Christoph Schlieder. Spatio-terminological information retrieval using the buster system. In Werner Pillmann and Klaus Tochtermann, editors, *EnviroInfo,* volume 1 of *Environmental Communication in the Information Society,* pages 93–100, Vienna, 2002. Berger Druck, Horn, Austria.
134. W3C. Date and time formats. Technical report, World Wide Web Consortium, 1998. W3C Note 27 August 1998, http://www.w3.org/TR/1998/NOTE-datetime-19980827, http://www.w3.org/TR/NOTE-datetime.

135. W3C. Extensible markup language (xml) 1.0 (second edition). Technical Report 6 October 2000, World Wide Web Consortium, October 2000. W3C Recommendation, http://www.w3.org/TR/2000/REC-xml-20001006,
136. W3C. Semantic web. Technical Report 17. October 2002, World Wide Web Consortium, 2000. http://www.w3.org/2001/sw/, no pagination, verified on Oct 17, 2002.
137. Holger Wache. Towards rule-based context transformation in mediators. In S. Conrad, W. Hasselbring, and G. Saake, editors, *International Workshop on Engineering Federated Information Systems (EFIS 99),* Kühlungsborn, Germany, 1999. Infix-Verlag.
138. Holger Wache. *Semantische Mediation für heterogene Informationsquellen,* volume 261 of *DISKI, Dissertationen zur Künstlichen Intelligenz.* Akademische Verlagsgesellschaft, Berlin, 2003.
139. Holger Wache, Thorsten Scholz, Helge Stieghahn, and B. König-Ries. An integration method for the specification of rule-oriented mediators. In Yahiko Kambayashi and Hiroki Takakura, editors, *International Symposium on Database Applications in Non-Traditional Environments (DANTE'99),* pages 109–112, Kyoto, Japan, 1999.
140. Holger Wache and Heiner Stuckenschmidt. Practical context transformation for information system interoperability. In V. Akman, P. Bouquet, R. Thomason, and R.A. Young, editors, *Modeling and Using Context,* volume 2116 of *Lecture notes in AI,* pages 367–380. Springer Verlag, Proceedings of the Third International and Interdisciplinary Conference, CONTEXT, Dundee, UK, 2001.
141. Holger Wache, Thomas Vögele, Ubbo Visser, Heiner Stuckenschmidt, Gerhard Schuster, Holger Neumann, and Sebastian Hübner. Ontology-based integration of information - a survey of existing approaches. In Asuncion Gómez Pérez, Michael Grüninger, Heiner Stuckenschmidt, and Mike Uschold, editors, *IJCAI-01 Workshop: Ontologies and Information Sharing,* pages 108–117, Seattle, WA, 2001.
142. A. N. Whitehead. *Process and Reality: Corrected edition.* Mac Millan Publications & Co, New York, 1978.
143. Gio Wiederhold. Mediators in the architecture of future information systems. *IEEE Computer,* 25(3):38–49, 1992. standard reference for mediators.
144. M. F. Worboys and P. Bofakos. A canonical model for a class of real spatial objects. In D. Abel and B. C. Ooi, editors, *Advances in Spatial Databases: 3rd International Symposium (SSD 93),* volume 692 of *Lecture Notes of Computer Science (LNCS),* pages 36–52. Springer Verlag, 1993.
145. M. F. Worboys and S. M. Deen. Semantic heterogeneity in distributed geographical databases. *SIGMOID Record,* 20(4), 1991.
146. Lofti A. Zadeh. Fuzzy sets. *Information and Control,* 8:338–353, 1965.